CHAPTER 7

DATA PROCESSING SUBSYSTEM

Data processing has rapidly become an integral part of Naval Security Group operations and is therefore of interest and importance to the maintenance technician. This chapter discusses the Data Processing Subsystem (see figure 3-1), its basic functions and components. A brief discussion of the absolute computer language (the binary base system), basic computer logic circuitry, and some of the terminology used in conjunction with the Data Processing Subsystem is presented so you can understand its inner workings. For more information on computer terminology see Appendix I Glossary.

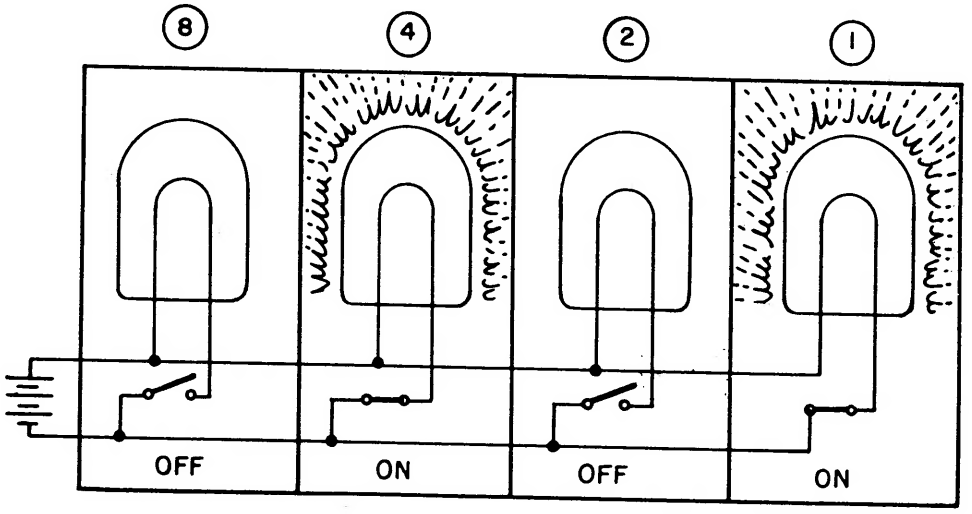
The Data Processing Subsystem topics discussed in this chapter are not meant to teach you to maintain a computer. The information contained herein is only basic in nature, for a more thorough knowledge the technician should study *Digital Computer Basics,* NAVTRA 10088-A.

ABSOLUTE COMPUTER LANGUAGE

BINARY NUMBER SYSTEM

The internal components of a computer (e.g., diodes, transistor flip-flops, and magnetic cores) that represent data can only represent two possible stable states, that of conducting or not conducting. The binary system, being base 2, is therefore a natural selection for use in a computer.

For example, we may represent decimal numbers (data) by using light bulbs which are assigned the decimal values of 8, 4, 2, and 1 reading from left-to-right in figure 7-1. A lighted



124.5

Figure 7-1.—Light Bulbs Assigned Decimal Numbers.

lamp in the 8's column represents a 1 condition in that column. The numerical value of this 1 condition is 8. A 1 condition at the 4's level represents 4, etc. The decimal value Ø is represented when all lights are off. The decimal value of 15 is represented when all lights are "on". The decimal value of 5 is represented by having the 4's light on and the 1's light on. The decimal value of 12 is represented by having the 8's light on and the 4's light on. Thus any decimal number from Ø to 15 may be represented by the four light bulbs.

The topic of number systems is discussed in detail in the rate training manual *Mathematics, Vol. 3*. In this chapter we will only review the main points of number systems as discussed in that course.

Basic Operations

Computation in the binary number system is relatively simple since that number system uses only the digits Ø and 1. You will note in the example below that the base of the number is signified by a subscript, e.g., $1ØØ1_2$, 11_8, $9_{1Ø}$.

ADDITION.–For addition three rules apply:

1. Zero plus zero equals zero.
2. Zero plus one equals one.
3. One plus one equals zero with a carry of one to the next position on the left.

Example:

Binary Representation	Decimal Equivalent
$ Ø1111_2$	$15_{1Ø}$
$+ ØØ111_2$	$+ 7_{1Ø}$
$1Ø11Ø_2$	$22_{1Ø}$

SUBTRACTION.–For subtraction four rules apply:

1. Zero minus zero equals zero.
2. One minus one equals zero.
3. One minus zero equals one.
4. Zero minus one equals one, with one borrowed from the left.

Example:

Binary Representation	Decimal Equivalent
$ Ø11Ø1_2$	$13_{1Ø}$
$- ØØ11Ø_2$	$- 6_{1Ø}$
$ØØ111_2$	$7_{1Ø}$

MULTIPLICATION.–For multiplication three rules apply:

1. Zero times zero equals zero.
2. Zero times one equals zero.
3. One times one equals one.

Example:

Binary Representation	Decimal Equivalent
$ Ø1111_2$	$15_{1Ø}$
$X 1ØØ_2$	$x4_{1Ø}$
ØØØØØ	$6Ø_{1Ø}$
ØØØØØ	
Ø1111	
$Ø1111ØØ_2$	

DIVISION.–For division similar concepts are applied:

Example:

Binary Representation	Decimal Equivalent
$1Ø1_2$	$5_{1Ø}$
$1ØØ_2 \overline{)1Ø1ØØ_2}$	$4_{1Ø} \overline{)2Ø_{1Ø}}$
1ØØ	2Ø
1ØØ	Ø
1ØØ	

OCTAL SYSTEM

The Octal System (base 8) is quite useful as an accessory to the binary system. Because eight is an integral power of two, it is an easy matter to convert from one system to the other. One octal digit is always equal to three binary digits and vice versa.

Octal to Binary	Binary to Octal
$2\ 7\ 5_8$	$\emptyset 1\emptyset\ 111\ 1\emptyset 1_2$
$\emptyset 1\emptyset\ 111\ 1\emptyset 1_2$	$2\ 7\ 5_8$

The use of octal numbers also reduces the number of digits required to represent the binary equivalent of a large decimal number.

The processes involved in the application of these binary/octal/decimal conversions are not presented in this text due to their complexity, however, figure 7-2 is a comparison table of the three most commonly used number systems. More information may be found in *Basic Electronics Vol. 2*, NAVPERS 10087-C, Chapter 13; and the NAVSEA/NAVSHIPS 0967-LP-000-0120, Electronic Circuits, Vol. II, Chapter 16 on the methods used to convert from one system to another.

BINARY	OCTAL	DECIMAL
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	6
111	7	7
1000	10	8
1001	11	9
1010	12	10
1011	13	11
1100	14	12
1101	15	13
1110	16	14
1111	17	15
10000	20	16

Figure 7-2.—Table, Comparing Number Systems Commonly Used in Digital Data Processing

BASIC COMPUTER CIRCUITRY

A majority of the circuits used in computers are unique and are not found in other types of electronic equipment. However, some of the circuits used in computers are also found in conventional electronic equipment. In practice, many variations of logic polarity are employed. A brief discussion of the two general classes of polarity is presented in the following paragraphs. Each of the logic circuit discussions also given in this section are based upon the form of logic polarity best suited for simple explanation of the circuit action.

POSITIVE LOGIC

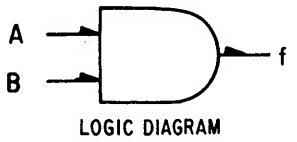
Positive logic polarity is defined as follows: When the logic 1(H) state has a relatively more positive electrical level than the logic $\emptyset$ (L) state, and the circuit is activated by the logic 1(H) signal, the logic polarity is considered to be positive.

The following typical examples illustrate the manner in which positive logic may be employed.

Example 1: Logic 1 = +1$\emptyset$ volts = H (high)
Logic $\emptyset$ = $\emptyset$ volts = L (low)

Example 2: Logic 1 = $\emptyset$ volts = H (high)
Logic $\emptyset$ = -1$\emptyset$ volts = L (low)

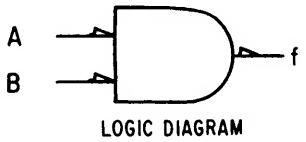
In both examples the logic 1 (H) state is always more positive than the logic $\emptyset$ (L) state, even though in example 2 the logic 0 (L) state is a negative voltage. When positive logic is used in any type of circuit, it is diagrammed by a solid right triangle, as indicated in Figures 7-3A and 7-4A, at the input and/or output of the logic device.



INPUTS		OUTPUTS
A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

TRUTH TABLE

A. POSITIVE AND GATE



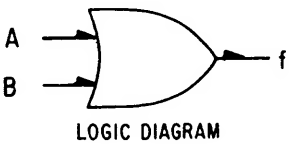
INPUTS		OUTPUTS
A	B	f
1	1	1
1	0	0
0	1	0
0	0	0

TRUTH TABLE

B. NEGATIVE AND GATE

1.451

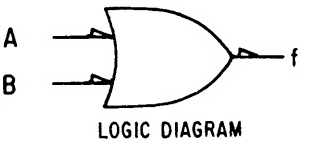
Figure 7-3.—Positive and Negative Logic AND Gates.



INPUTS		OUTPUTS
A	B	f
0	0	0
0	1	1
1	0	1
1	1	1

TRUTH TABLE

A. POSITIVE OR GATE



INPUTS		OUTPUTS
A	B	f
1	1	1
1	0	1
0	1	1
0	0	0

TRUTH TABLE

B. NEGATIVE OR GATE

1.452

Figure 7-4.—Positive and Negative Logic OR Gates.

NEGATIVE LOGIC

Negative logic polarity is defined as follows: When the logic 1(L) state has a relatively more negative electrical level then the logic Ø(H) state, and the circuit is activated by the logic 1 (L) signal, the logic polarity is considered to be negative. The following typical examples illustrate the manner in which negative logic can be employed.

Example 1: Logic 1 = Ø volts = L (low)
Logic Ø = +1Ø volts = H (high)

Example 2: Logic 1 = -1Ø volts = L (low)
Logic Ø = Ø volts = H (high)

In both examples the logic 1 (L) state is always more negative than the logic Ø (H) state, even though in example 1 both states are in the positive region. When negative logic is used in any type of logic circuit, it is diagrammed as an open right triangle, as indicated in figures 7-3B and 7-4B, at the inputs and/or outputs of the logic device.

NOTE: In both, positive and negative logic polarity, a high (H) logic state is always the state in which the normal input voltage level is the more relatively positive level.

LOGIC GATES

One of the most important and necessary logic elements of a data processing device is the gating circuit. A gate is defined as an electronic circuit employed as a logic device, designed to yield a change in its output condition when definite predetermined conditions are present at its input terminals. Inputs to the gates may number two or more and may be levels, pulses, or both. These versatile circuits find application as logic AND, OR, NOR, and NAND gates and as certain special combinations and applications. Their use depends upon the output desired for a

particular application. Some types of gates provide inversion while others do not.

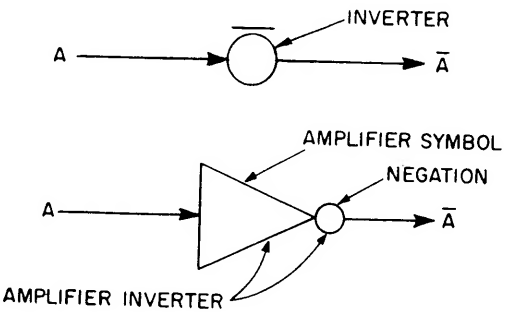
AND GATES.–An AND gate (figure 7-3), used in logic networks, is one whose output is a logic 1 only when all inputs are logic 1's. The output of an AND gate is always a logic Ø anytime a logic Ø appears on any of the input terminals. The truth tables shown in figure 7-3 illustrate the operating conditions of two typical AND gates. Diodes, transistors, vacuum tubes, resistors, and resistor capacitor circuits are used as AND gates. The symbol for an AND gate (figure 7-3) merely indicates its logic functions and does not indicate the electronic circuit type.

OR GATES.–An OR gate (figure 7-4) is a gate which produces a logic 1 output when any one or more of the inputs are a logic 1. The output is a logic Ø only if all inputs are logic Ø's.

NOT FUNCTION (INVERTER OR NEGATION).–The basic NOT function is a single input circuit (figure 7-5) arranged so the output is always the inverse of the input. That is, if the input is a "high" the output will be a "low" and vice versa. This function, also called an inverter or negation, is represented by a circle on the input and/or output of other logic symbols such as NOR and NAND (discussed below).

NOR GATES.–The OR gate with inversion (negation) is sometimes called NOR (NOT OR). A truth table and the logic symbol for the NOR gate is illustrated in figure 7-6. The OR gate with inversion is distinguished from the OR gate previously discussed by the presence of polarity inversion, as shown on the output of figure 7-6B. The basic OR has no inverter on the output (see figure 7-4).

NAND GATES.–The AND gate with inversion (negation) is sometimes called NAND (NOT AND), and is shown in figure 7-7. The AND gate with inversion is distinguished from the AND gate previously discussed by the presence of polarity inversion, as shown on the output of figure 7-7B. The basic AND has no inverter (see figure 7-3).



A

TRUTH TABLE

A	$\overline{A}$
0	1
1	0

B

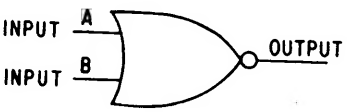
1.453

Figure 7-5.–The NOT Function.

TRUTH TABLE

A	B	OR OUTPUT	NOR OUTPUT
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

A



B

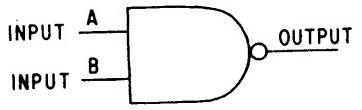
1.454

Figure 7-6.–The NOR Gate.

TRUTH TABLE

A	B	AND OUTPUT	NAND OUTPUT
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

A



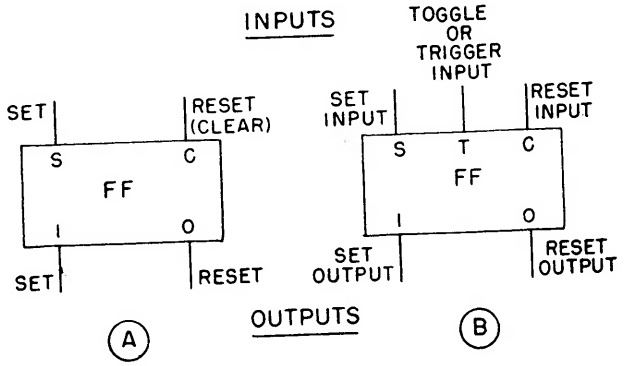
B

1.455

Figure 7-7.—The NAND Gate.

FLIP-FLOP CIRCUITS

The FLIP-FLOP is basically a bistable multivibrator. Figure 7-8A illustrates the basic FLIP-FLOP symbol. The circuit has two inputs called the SET and RESET inputs.



124.26

Figure 7-8.—Flip-Flop Symbol.

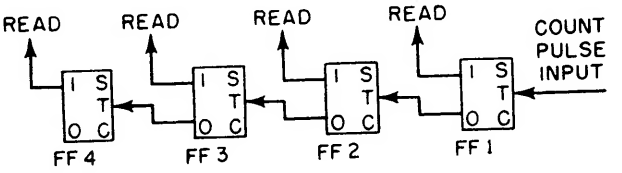
A "1" input to the SET side produces a "1" at the SET(s) output. Once in the SET state, the output will remain the same until a subsequent "1" input to the RESET side causes the FLIP-FLOP to change states. This produces a "1" output on the RESET (C) or CLEAR output side, and a "Ø" on the SET(s) side.

The TOGGLE or TRIGGER (T) input (figure 7-8B) triggers both inputs simultaneously causing the FLIP-FLOP to reverse states; provided the proper steerage is present, i.e., if the FLIP-FLOP is set, a high must be present at the CLEAR input for the trigger to cause a change in state.

FLIP-FLOPs are used in counters and other types of circuits which will be described later.

Counter Circuits

The most common counter circuit found in computers is the binary counter. A binary counter is merely a series of flip-flops. The number of flip-flops used determines the modulus (fixed counting capacity plus one) of the counter. Figure 7-9 shows a four-stage binary counter. The input pulses are applied to the flip-flop corresponding to the lowest order of binary number. Note that the reset output from the "Ø" side of each flip-flop feeds the toggle (T) input of the next (higher order) flip-flop and the set output provides the count readout. If a flip-flop is in the set position, the output from the "1" side of the circuit reads 1; if the circuit is in the reset position, the "1" side output reads Ø. Every time the reset (clear) side changes from a Ø to 1 state (1 to a Ø on the set side), a pulse is sent to the next higher order



164.21

Figure 7-9.—Four-Stage Binary Counter.

flip-flop input toggle. An input to the counter, shown in figure 7-9, affects the flip-flops in the following manner:

Step 1. Assume that all four flip-flops are in the reset position. Each output reads Ø and the total counter output is $ØØØØ_2$ (binary Ø) which is the first counting number.

Step 2. A single pulse is introduced. FF1 changes to its set position, producing a 1 at its output. However, the signal is not coupled to FF2 because the flip-flop produces a pulse at its reset output only when changing from set to reset. The counter's output now reads $ØØØ1_2$ (binary 1).

Step 3. Another pulse enters the counter's input. FF1 changes from set to reset, producing a Ø at its set output. This time, a pulse is produced at the reset output (reset side changes from Ø to 1). The reset output pulse is coupled to FF2, causing it to change from the reset to the set position. The FF2 set output now reads 1. (Again, because each flip-flop couples a pulse only when going from the set to reset position, no pulse is produced from the FF2 reset output, and the remaining flip-flops are not affected). The counter's output now reads $ØØ1Ø_2$ (binary 2).

Each successive input pulse to the counter affects the flip-flops in a similar manner. The third input pulse will change FF1 to the set stage, producing $ØØ11_2$ (binary 3). The fourth pulse will change FF1 and FF2 to the reset position and FF3 to the set position, producing $Ø1ØØ_2$ (binary 4). Continuing this analysis reveals that the circuit will count up to 15 pulses, indicating its tally in binary code.

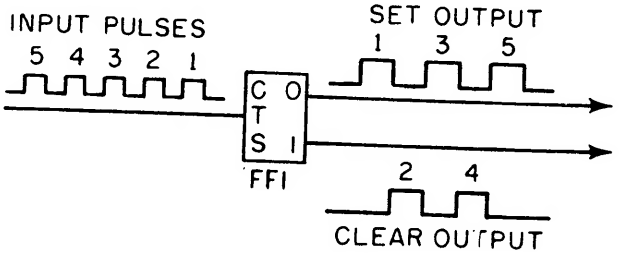
Thus, the highest possible binary number in the four-stage counter's output is 1111_2 (decimal 15). The modulus, therefore, is 15 plus the $ØØØØ_2$ count or 16. Since the modulus is 16, the addition of another flip-flop will permit counting to 32.

Binary counters are also used to count the number of input pulses to the circuit and produce an output after a certain number of pulses are received. In this way the counter divides the input frequency of pulses by some factor, and is therefore referred to as a frequency divider. The function of a binary frequency divider or counter, is to divide, or scale down, the number of input pulses by some power of two. A single flip-flop, for example, divides by two. It produces one output pulse for every two input pulses.

The output pulses can be either in phase or out-of-phase with the pulses to be counted. The input and output pulses shown in figure 7-10 are positive. Flip-flop FF1 can be made to produce a positive output so that a second flip-flop of the same type can, upon receiving the output of the first, again divide the input by 2. Thus, the output of a two stage circuit produces only one pulse for every four pulses at the input, making it a divide-by-four circuit. Similarly, a third flip-flop will make it a scale-of-eight divider, producing one output pulse for every eight input pulses. Figure 7-11 shows a four-stage binary scaler and the pulse outputs of each stage. Note that each flip-flop produces only half as many pulses as the one before it, so that the final stage (FF8) produces only one pulse for every 16 pulses introduced at the first stage. The circuit is therefore a scale-of-16 divider.

Another type of counter is the ring counter. The ring counter is cyclic, having no beginning and no ending. After the counter has reached its highest possible count, its next count is Ø.

In the ring counter, only one element or stage is on at any one time. Each input signal



164.23

Figure 7-10.—Using a Flip-Flop as a Scale-of-Two Divider.

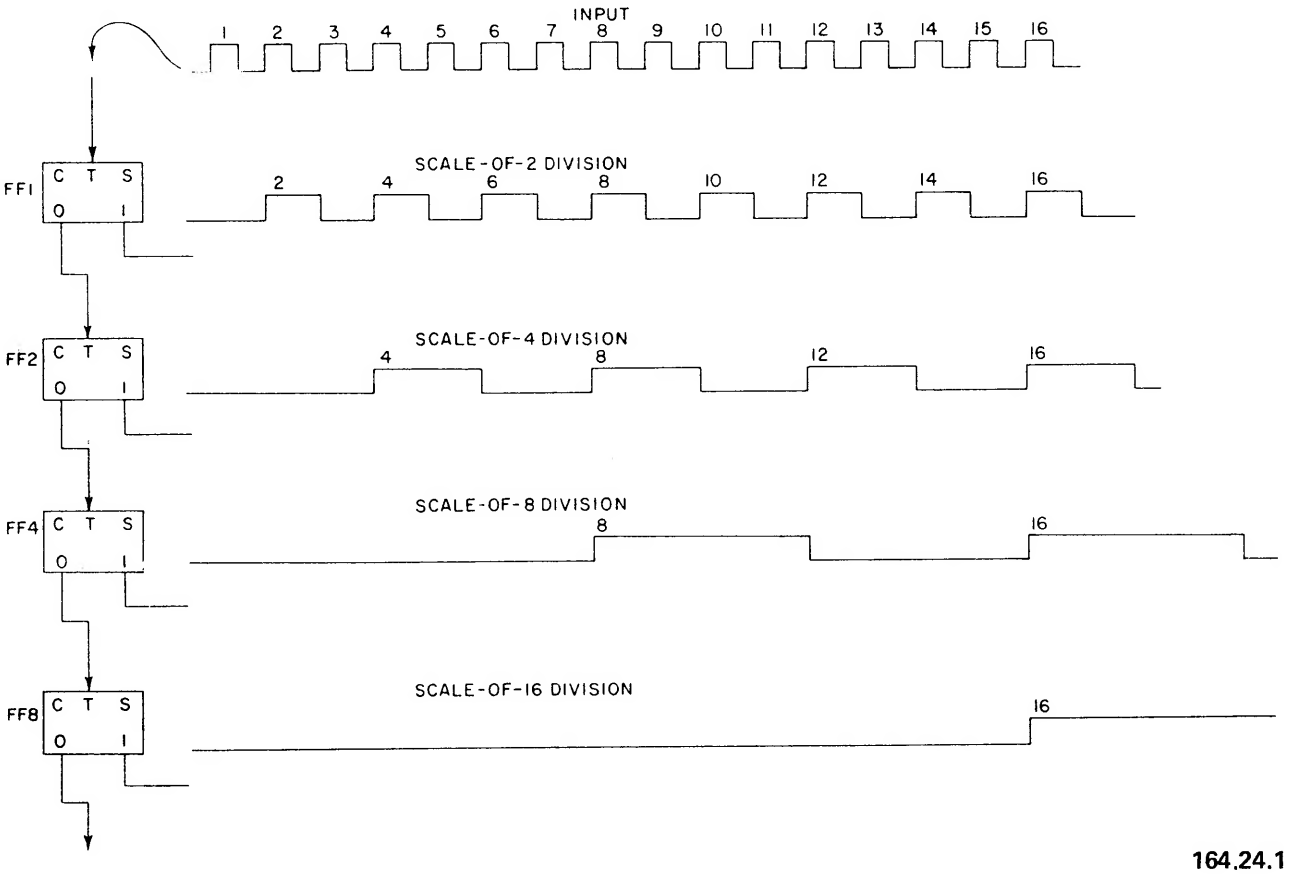


Figure 7-11.—Pulse Train of a Four-Stage Binary Scaler.

advances the ON or operating stage one step forward.

The ring counter shown in figure 7-12 uses flip-flops and AND circuits. Note that the input pulses never enter any flip-flop directly. Instead, they are applied to all of the AND circuits. An input pulse is used by each AND circuit to clear the flip-flop corresponding to that AND circuit (if the flip-flop is in the ON state), and to set the following flip-flop.

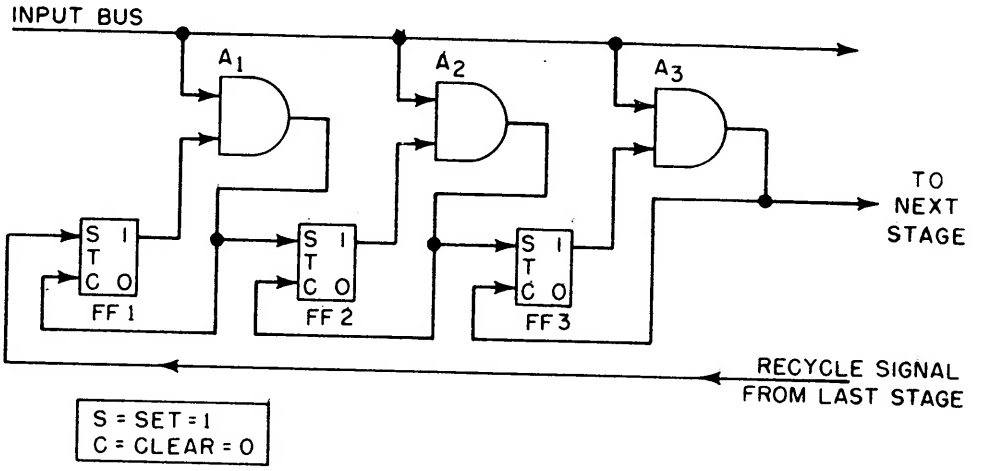
Note that the connection between flip-flops FF2 and FF3 are identical to the connections between FF1 through the last flip-flop and back to FF1.

Thus, it can be seen that the counter uses a cyclic principle, and that only one stage is ON at any one count. The modulus of a ring counter can be increased by adding more elements (flip-flops and AND circuits in this case).

Registers

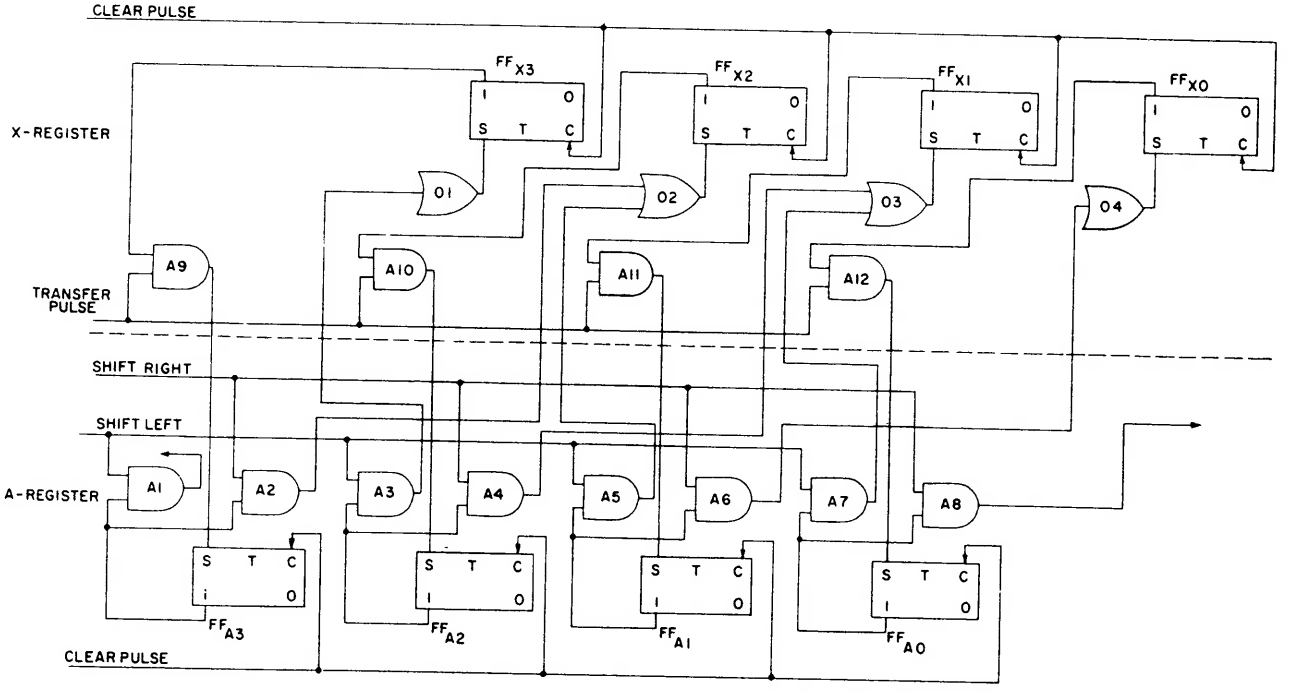
A register is used as a temporary storage device in a computer. Data is put into a register and can be read out of the register when needed for some operation. Shift registers may use either serial or parallel input or output technique. A data word may be read into a register serially and read out in parallel, according to design. Shift registers can shift the data bits to the right or to the left, round off a number, or multiply or divide by 2, 4, 8, 16, etc. Figure 7-13 is a shift register which illustrates the concepts of right and left shifting using parallel techniques.

A discussion of the shift register operation will not be presented here due to its complexity,



164.22

Figure 7-12.—Ring Counter Using Flip-Flops and AND Circuits.



164.25

Figure 7-13.—Shift Register in Arithmetic Section.

however, a good description of its operation is presented in *Digital Computer Basics,* (NAVTRA 10088).

COMPUTERS

The computer has many applications within the Naval Security Group. Being a very flexible and versatile tool, the computer is invaluable in the areas of administration, management, and operations. Its functions include personnel accounting, data retention, 3-M system planning, updating, maintenance schedule listings, equipment inventories, budgetary control, and various other purposes.

The main use of the computer in Naval Security Group operations is for rapid acquisition and processing of data. Computers located at HFDF net control stations provide a highly automated system for net coordination and control. In addition, the computers located at net control stations perform computations (i.e., standard deviation, systematic error, ionospheric propagation, etc.) to furnish fix data on target signals.

CAPABILITIES

The two major characteristics of computers which make them so useful in military and commercial applications are SPEED and ACCURACY. The speed of computers is apparent when we consider that problems which once required days, weeks, or years to solve by manual methods, can be solved in seconds or minutes by a computer. The speed of computers is even more apparent when one considers that a single arithmetic operation can now be solved and stored by a computer in less than a microsecond, whereas a mathematician needs a few seconds to do the same operation and record (or store) it on paper. Thus, the computer can solve the problems and produce an output record of its results, thousands or even millions of times faster then man.

The second characteristic of the computer is ACCURACY. Once a computer is provided with the correct instructions, the planned operations can be repeated millions of times without a single error. Computers make errors only when there is a breakdown in the computing system, or when there is a human error in the prepared instructions. Once the breakdown or error is detected and corrected, the computer again operates at high speed and without errors.

TYPES OF COMPUTERS

Computers are classified into two general types; digital and analog, although there are variations of these types called "hybrid computers," which have both digital and analog characteristics.

Analog Computers

The analog computer accepts continuously varying inputs, and supplies an instantaneous continuously varying output. Analog computers use physical changes as input data and indicates the significance which such changes have on the device or unit as a whole. The input/output data may be either electrical, mechanical, or a combination of both. Conditions such as temperature, pressure, and angular position, must be represented by electromechanical analogies. From this, note that an analog computer is, by nature, limited, in its application to problems related to specific devices.

The accuracy of the analog computer is limited by the precision of its components, i.e., potentionmeters, resistors, etc. Thus, accuracies greater than approximately .001% are not economically feasible.

Digital Computers

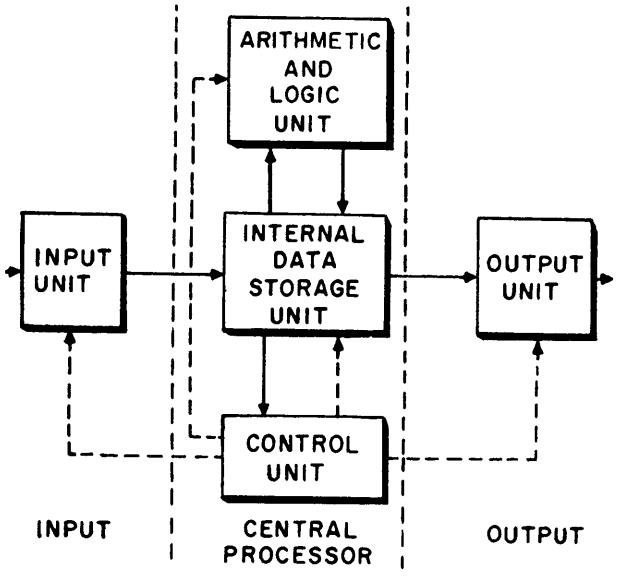
The digital computer, as implied by its name, produces an output by responding to changes in fixed increments, such as 0 to 1, or 1 to 2. The change may be accomplished in a gear train, by changing a voltage from one level to another, by the "on" or "off" condition of a switch, by the energized or deenergized condition of a relay, or by the presence or absence of electrical pulses.

In this text we shall concern ourselves with electronic digital computers. Here too, the input data must be in discreet increments (digital form). Three factors determine the rate at which the output data is available. These are: memory access time, instruction execution time, and the complexity of the problem.

Electronic digital computers can be made accurate to any desirable degree. They are usually more expensive than analog computers but are also usually more versatile. A digital computer can be given a sequence of instructions in which it can execute later steps using the results of the earlier steps. It can also alter the sequence of instructions according to the results of previous steps.

BASIC DIGITAL COMPUTER DIAGRAM

The basic sections or units of a digital computer are shown in figure 7-14. The three center blocks (arithmetic and logic, internal data storage, and control units) comprise what is generally referred to as the "central data processor." Each of these basic units will be discussed separately in the following paragraphs.



12.136

Figure 7-14.–Representative Digital Computer Block Diagram.

Control Unit

The control section is comparable to a telephone exchange. It directs the operations of the computer under the direct influence of a sequence of instructions called the "program." The instructions are comparable to the phone numbers dialed into a telephone exchange and cause certain switches and control lines to be energized.

The program may be stored in the internal circuits of the computer or it may be read instruction-by-instruction from external media. The internally stored program type of computer, generally referred to only as a "stored program" computer, is the most pratical type to use when speed and fully automatic operation are desired.

In addition to the command which tells the computer what to do, the control unit also dictates how and when each specific operation is to be performed. It is also active in initiating circuits which locate any information stored in the computer and in moving this information to the point where the actual manipulation or modification is to be accomplished.

In the stored program computer, the control unit reads an instruction from the memory section (as instructed by the program). The information read into the control unit from memory is in the form of voltage levels that make up a "binary word," and represents a specific operation that is to be performed. The location of the data to be operated on is generally a part of the instruction, and energizes circuitry which causes the specified operation (add, subtract, compare, etc.) to be executed. Subsequently, the control unit reads the next instruction or jumps as directed, to find the next instruction to execute.

The four major types of instructions are: (1) transfer; (2) arithmetic; (3) logic; and (4) control. Transfer commands are those whose basic function is to transfer data from one location to another. One of the locations is an

address in memory and the other is either a register or an input/output device. Arithmetic instructions are those which combine two pieces of data to form a single piece of data using one of the arithmetic operations. In some types of computers, one of the pieces of data is in a location specified by the address contained in an instruction, and the other is already in a register (usually the accumulator). The results are usually left in the accumulator.

Logic instructions make the digital computer into a system which is more than a high speed adding machine. By using logic instructions the programmer may instruct the system on various alternate sequences through the program. For example, through the use of logic instructions, a computer being used for maintenance inventory will have one sequence to follow if the number of a given item on hand is greater than the order amount and another sequence to follow if the number on hand is smaller than the order amount. The choice of which sequence to use will be made by the control unit under the influence of the logic instruction. Logic instructions provide the computer with the ability to make decisions based on the result of previously generated data.

Control instructions are those which are used to send commands to devices which are not under direct command of the control unit, such as input/output units. The address contained in the instruction does not specify a location in memory but is usually a code group specifying an action required of a particular piece of equipment.

In a single address compute, i.e., where each instruction refers to only one address or operand, the instructions are normally taken from the memory in sequential order. If one instruction comes from a certain location, say X, the next instruction is usually taken from location X + 1. However, the execution of a logic instruction may produce a result which dictates that the next instruction is to be taken from an address as specified in a portion of the logic instruction. For example, the logic instruction may cause certain operations in the computer to determine if the content of a given register in the arithmetic section is negative. If the answer is "yes", the location of the next instruction is that specified in an address section of the logic instruction. If the answer is "no", the next instruction would be taken from the next sequential location in the memory.

Every computer provides circuitry for a variety of logic instructions for choosing alternate instruction sequences if certain desirable or undesirable conditions exist. The ability to "branch" at key points is the special feature of the computer that makes it able to perform such diverse tasks as missile control, accounting, or tactical air plotting.

Memory Unit

In most digital computers the internal data storage or memory section is constructed of small magnetic cores, each capable of representing an "ON" ("1") or "OFF" ("Ø") condition. A system of these cores arranged in a matrix can store any computer word which is represented in binary form.

All computers must contain facilities to store computer words or instructions (which are intelligible to the computer) until these instructions or words are needed in the performance of the computer calculations. Before the stored program type computer can begin to operate on its input data, it is first necessary to store, in memory, a sequence of instructions and all figures, numbers, and any other data which are to be used in the calculations. The process by which these instructions and data are read into the computer is called "loading."

Actually the first step in loading instructions and data into a computer is to manually place enough instructions into memory by using the console or keyboard so that these instructions can be used to bring in more instructions as desired. In this manner a few instructions are used to "bootstrap" more instructions. Some computers make use of an auxiliary (wired) memory which permanently stores the "bootstrap program," thereby making manual loading unnecessary.

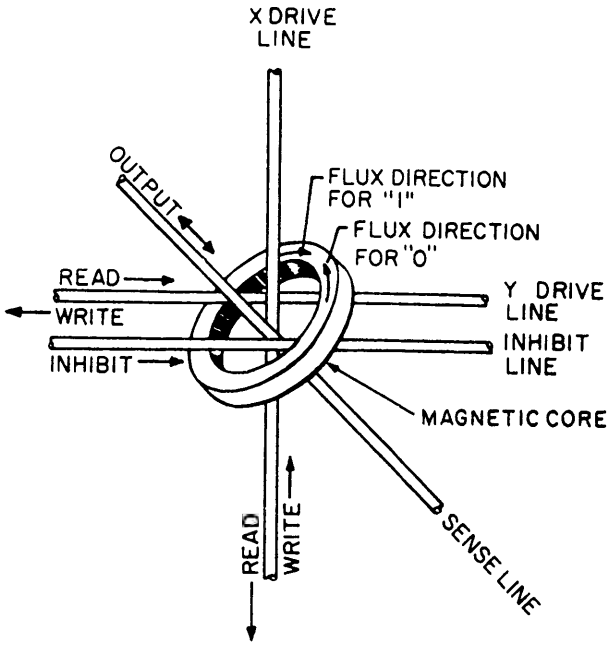
The memory (or storage) section of a computer is essentially an electronically operated file cabinet. It is actually a large number (generally between 1 and 4Ø thousand) of storage locations; each referred to as a storage address or register. Every computer word which is read into the computer during the loading process is stored or filed in a specific storage address and is almost instantly accessible.

STORAGE DEVICES.—The types of memory storage devices most frequently used in present day computer technology are magnetic drums, magnetic cores, thin film, magnetic disks and magnetic tapes.

Magnetic Cores.—One of the most advanced methods to date for storing internal data in a computer is realized by using magnetic cores. Cores are generally constructed by either one of two methods: the first type of core, called a tape-wound core, is fabricated by wrapping a tape of magnetic material around a nonmagnetic toroidal form. A toroid is a term used to describe a doughnut-shaped solid object. The second type of core is called a ferrite core, and it is made by molding finely ground ferrite into a toroidal form. The ferrite used in this application is a ceramic iron oxide possessing magnetic properties. The ferrite particles are then heat fused or "sintered" by the application of heat and pressure.

In magnetic core memories, each data bit is stored in the magnetic field of a small, ring-shaped magnetic core (figure 7-15). Magnetic cores generally have four wires running through them. Two wires are used for Read selection. (These same two wires are used for write by reversing the direction of current flow.) An inhibit wire prevents writing a "1" when a "Ø" is to be written and the sense wire picks up the signal voltage generated by the shifting of a core from "1" to "Ø" in a read cycle.

Since a single core stores only one bit of a word, a large number of cores are required to handle all the bits in every word to be stored. These cores are arranged in "arrays," similar to the tracks and slots used in magnetic drum storage, in order to assign memory address



124.56

Figure 7-15.—Magnetic Core Showing "X," "Y," Inhibit, and Sense Lines.

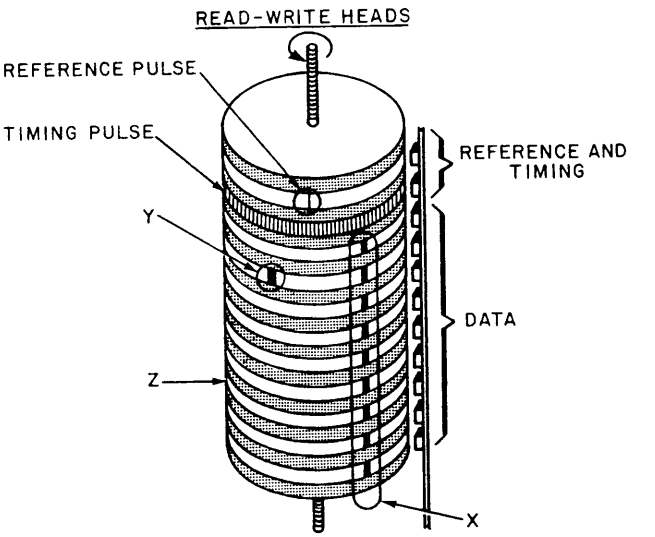
locations and quickly write data and locate data for read-out purposes. The technique used most frequently for writing and reading data in magnetic core arrays is known as the coincident-current technique. This technique is explained in subsequent paragraphs.

In computer memory applications the ferrite core is magnetized by a flux field produced when a current flows in a wire (drive line) that is threaded through the core. It retains a large amount of this flux when the current is removed. Flux lines can be established clockwise or counterclockwise around the core depending upon the direction of the magnetizing current. A current in one direction establishes a magnetization in a core in a given direction. Reversing the direction of the current flow reverses the direction of the flux field and the core magnetization. These two unique states represent "Ø" and "1", respectively.

Thin Film.—Thin film memory consists of a ferromagnetic material, permalloy, deposited

(under controlled conditions in a vacuum chamber) on a supporting material (substrate) of thin glass. After all air has been removed from the chamber, a shutter arrangement is opened and vapors from molten permalloy pass through a mask and are deposited on the supporting material (substrate). The pattern thus formed is determined by the shape of the mask. The thickness of each spot (magnetized area) is controlled by the amount of time the shutter is open.

A magnetic field is applied parallel to the surface of the substrate during deposition. The film spots thus become easier to magnetize in a direction parallel to that in which the magnetic field was applied during the deposition process. This direction is known as the preferred direction, likewise the axis of this magnetism is referred to as the preferred axis.



124.60

Figure 7-16.—Magnetic Drum.

Magnetic Drums.—The magnetic drum storage device is a cylinder which rotates at constant velocity. Information is written on or read from the drum when its magnetic surface passes under magnetic heads which are similar to the magnetic heads found on commercial tape recorders.

Magnetic drums provide a relatively inexpensive method of storing large amounts of data. A magnetic drum (figure 7-16) can be made by using either a hollow cylinder (thus the name "drum") or a solid cylinder. The cylinder may consist entirely of a magnetic alloy, or it may have such an alloy plated upon its surface. Many drums are made by spraying on a magnetite, such as iron oxide. The surface is then coated with a thin coat of lacquer, and buffed.

Representative drums have diameters ranging from 12.7 to 5Ø.8 centimeters (5 to 2Ø inches). The surface of the drum is divided into tracks or channels which encircle the drum. A number of read and write heads (at least one for each track) are used for recording and reading. The drum is rotated so that the heads are near but not touching the drum surface at all times.

As the drum rotates, the tracks are continuously passing under their respective head. Each track is subdivided into cells, each of which can store one binary bit. All of the cells that are positioned under the heads of a multitrack drum at the same time are called a "slot." With some drums, each head reads or writes one bit of a word. Thus, when a word is written into or read from a slot, each track contains one bit of that word. The number of heads used depends on the size of the word that the computer is designed to handle.

One of the tracks provides timing signals for the drum rotation. The timing track determines the location of each set of storage cells around the drum. Each timing signal denotes a unit of time of the drum rotation. For example, if the timing track is 8Ø inches long and timing signals are recorded at 12Ø pulses per inch, there are 96ØØ locations for bit storage on the track. If the drum has 32 tracks in addition to the timing track, the drum has the capacity to store a total of 3Ø7,2ØØ bits.

Some drums use two or even three timing tracks. The timing tracks are used for synchronization purposes, and are sometimes called "control" or "clock" tracks. The timing pulses establish the time scale to which all circuits throughout the computer are synchronized.

When core memory is used all the data is stored in the cores in a static condition, and can be located at a given place at any instant and, easily read from that location in serial or parallel form to represent the same data that was stored in that location.

Dynamic storage refers to data stored on constantly rotating drums. Therefore, transfer of the data is complicated. Timing pulses are not used to synchronize the drum speed (which may vary slightly from time to time). Thus, some method must be used to ensure that data read into the drum memory in a given bit position, will be read from the memory with the same time reference. The probability of an incompatible time relationship between the drum speed and synchronizing (clock) pulses, makes it necessary to establish some means of compensating for variations in drum speeds.

In practice, the drum contains a control point and a number of sectors in a specific format. The control point is a magnetic mark that specifies a starting location on the drum. All data stored on the drum are referenced to this indexing point or "reference pulse" (shown in figure 7-16).

Magnetic Tapes.–Another form of memory is magnetic tape. Magnetic tape is widely used as a storage medium for large amounts of data. It is normally not used as an internal (main) storage medium because of its long access time. This can be readily realized if one considers that needed information is widely (and sometimes randomly) distributed along the tape. Thus, the two main advantages to be gained by using magnetic tapes are its large storage capacity and low cost.

Magnetic Discs.–The magnetic disk is a convenient medium for semipermanent storage of mass volumes of production programs. For many applications, disks are superior to magnetic tape for rapid acquisition and storage of mass volumes of systems programs and data.

Magnetic disks resemble phonograph records which have been coated with iron-oxide. The disks (or records) are arranged in stacks in much the same way as a record stack in a modern "juke box." All of the disks are continuously revolving and spaced apart so that a record head driven by an access mechanism can be positioned between the disks.

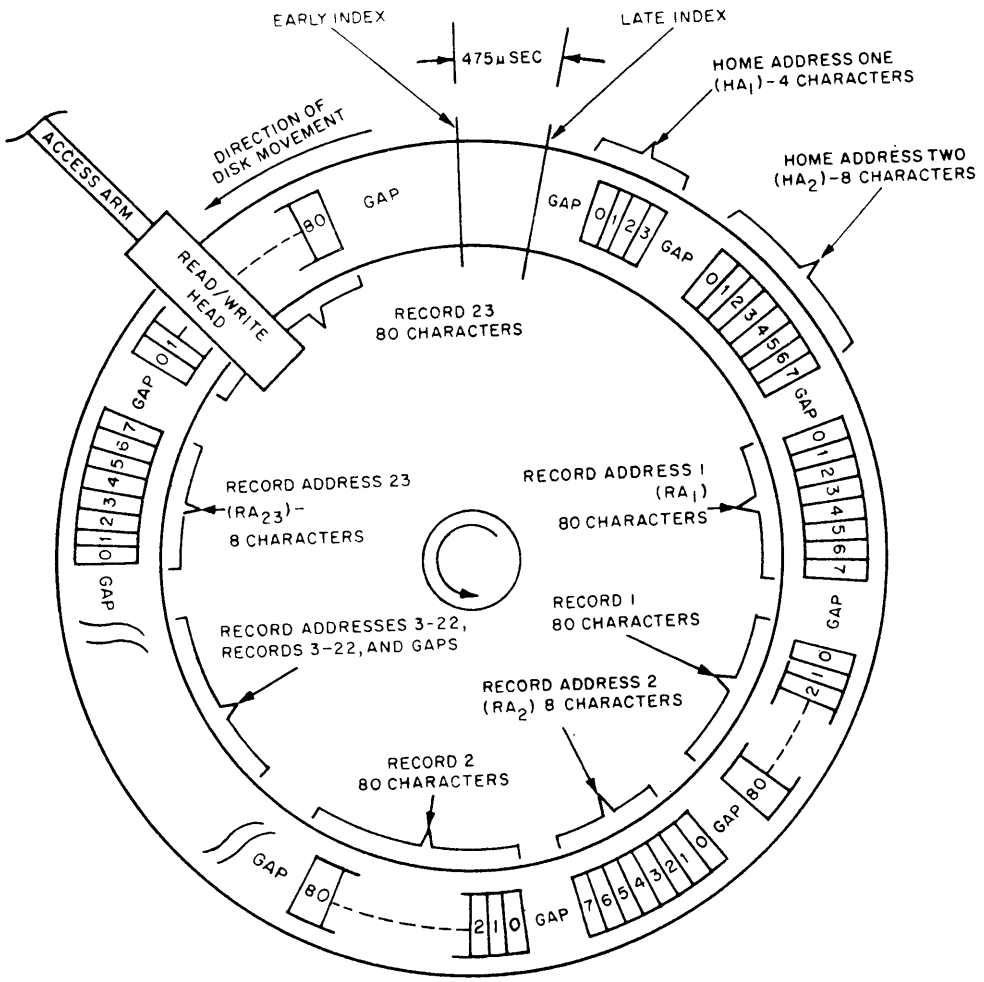
Data is recorded at a certain address on a specified disk. When readout of a particular bit of data is desired, the recording head is automatically positioned and the data is read serially from the surface of the selected disk.

The basic unit of information on the disk is a character. By design each character contains a given number of bits (for fixed word applications). One or more of these characters in a group form a record. A circular data track (figure 7-17) consists of one or more records, associated record addresses, gaps, and data track identification. A track of information as shown in this figure is defined in more detail later. A number of data tracks aligned on vertically arranged disks (figure 7-18) form a cylinder of information. A magnetic disk file system may contain one or more bands (modules). Each module contains a specified number of disks with their associated cylinders and data tracks. The flowchart in figure 7-19 illustrates the procedures necessary to retrieve or store information.

Arithmetic Unit

The arithmetic unit of the computer is the section in which arithmetic and logic operations are performed on the input or stored data. The arithmetic operations performed in this unit include adding, subtracting, multiplying, dividing, counting, shifting, complementing, and comparing.

Generally information delivered to the control unit represents instructions, whereas information routed to the arithmetic unit represents data. Frequently it is necessary to modify an instruction. This instruction may have been used in one form in one step of the program but must be altered for a subsequent step. In such cases, the instruction is delivered to the arithmetic unit where it is altered by addition-to or subtraction-from another number in the accumulator. The resultant modified



164.43

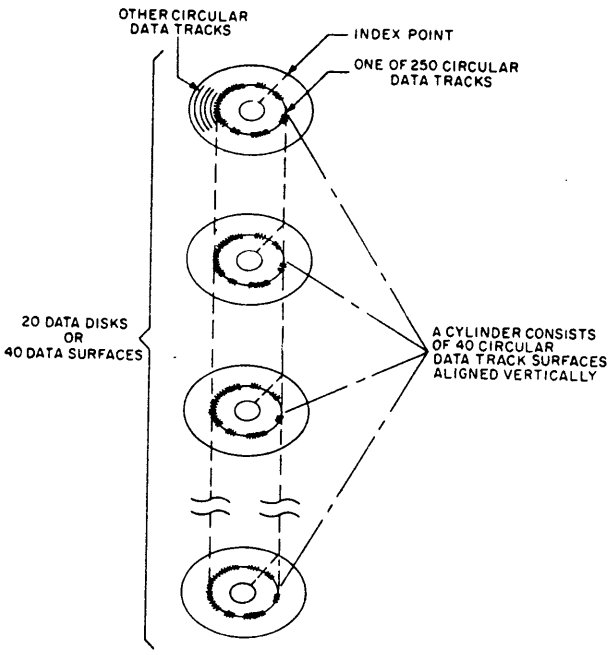
Figure 7-17.—Circular Data Track.

instruction is again stored in the memory unit for use later in the program.

All arithmetic operations can be reduced to any one of four arithmetic processes; addition, subtraction, multiplication, or division. In most computers, multiplication involves a series of additions; and division, a series of subtractions.

The arithmetic unit contains several registers; units which can store one "word" of computer data. This group of registers generally include D-, X-, and Q-registers (so named for identification purposes only), and a unit called an "accumulator" (A-register). During an arithmetic process, the D-, X-, and Q-registers temporarily hold or store the numbers being used in the operation, called "operands". The accumulator stores the result of the operation. The control unit instructs the arithmetic unit to perform the specified arithmetic operation (as requested in the instruction); transfers the necessary information into the D-, X-, and Q-registers from memory; and controls the storage of the results in the accumulator or in some specific location in memory.

The arithmetic unit also makes comparisons and produces "yes" or "no" or "go-no-go" outputs as a result. The computer may be



164.44

Figure 7-18.—Data Storage Disk Assembly.

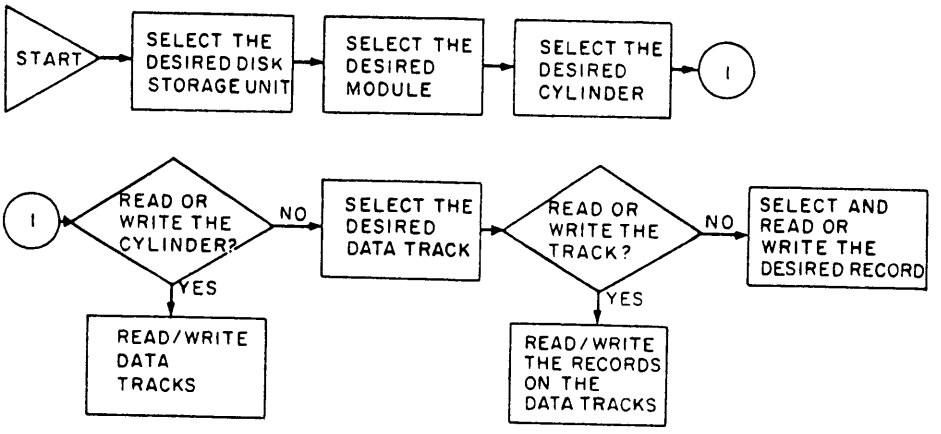
programmed so that a "yes" or "go" result advances the computer units to perform the next step in the program, whereas a "no" or "no-go" instruction may cause the computer to jump several programmed steps. A computer may also be programmed so that a "no" result at a certain point in the program will cause the computer to stop and await instructions from a keyboard or other input device.

INPUT/OUTPUT DEVICES

Input and output devices provide the computer with the facilities necessary for communicating with the users. Input devices such as consoles, card readers, tape readers, and typewriters supply the computer with data and instructions, while output devices provide the means for changing the data processed by the computer into a form specified by or intelligible to the users. The selection of input/output devices depend on the specific use for which a computer is intended. In the following discussions, the applications, types, and characteristics of input/output devices are presented.

GENERAL REQUIREMENTS FOR INPUT/OUTPUT DEVICES

Generally, input/output devices must meet two basic requirements. First, the devices must be able to modify all data so that it is acceptable to the computer during the input phase of the operation and must be able to present data in



164.45

Figure 7-19.—Flowchart for Storage and Retrieval of Data from Disks.

usable form during the output phase. Second, the devices must operate quickly and efficiently in conjunction with the computer.

Conventional input devices read coded data into computers or other devices from punched cards or tape (using holes punched in various positions of the cards or tape to represent data), or from magnetic tape (using magnetized areas on the tape). In some special military applications, the computer input is received from special purpose devices, such as radar sets, gun platforms, missile guidance systems, or tactical display consoles. In scientific digital computers, the input device may consist of a keyboard, while the output device may consist of a plotting board or an electric typewriter.

Data may be presented at the output in printed form (English or numerals), in plotted form (such as maps and graphs), on punched cards or tapes, magnetic tapes, or oscilloscopic displays. Outputs in still other forms are available for special applications.

Nearly all input/output devices suffer the same disadvantages—slowness of response. Most computers can process millions of bits of data per second. Input/output devices, particularly those which require some mechanical operation, are hard-pressed to manipulate several thousand bits per second. There is, of course, a wide disparity in data exchange rates between the various devices, but the computer is faster even when the fastest of these devices are considered.

Various procedures are being used in order to more profitably utilize computer time. One such procedure, designed to minimize computer idling time, is to program I/O cycles to run concurrently with computation. Another is to use a number of I/O channels and provide multiplexed inputs to the computer from several I/O devices. (Multiplexing, as used here, refers to the ability of the computer to sample the data on a number of input channels while maintaining the intelligence of the data from each channel. The rate of the multiplexing action is high enough to permit the operator of each I/O device to retain immediate access with the computer.)

Common methods for improving computer usage in so far as can be accomplished by the I/O equipments themselves include the use of off-line devices (i.e., devices not under the direct control of the computer), use of electronic switches to multiplex several equipments on one channel, and the use of buffer storage registers in the I/O equipment.

Buffer Storage

Buffers serve as intermediate storage devices to facilitate transfer of data between two mediums whose operating speeds are difficult, or impossible, to synchronize. It is frequently necessary to read data from cards, punched tapes, keyboards, etc., into the main (primary) computer memory. In most cases, the speed at which input devices can supply data to storage cannot be increased sufficiently to match the ability of the computer to read-in the data at electronic speeds. The same incompatibility is encountered when reading data from memory to output devices. A buffer device is therefore designed to read-in or write-out data at speeds which are compatible with both the input/output devices and the main computer memory.

Several types of buffers are in use. The simplest type is an arrangement of flip-flop registers into which data can be slowly accumulated but can be released or read-out at electronic speeds. By design, data can be read-into and out-of the registers in either serial or parallel form. The buffer storage must be capable of reading data slowly from the input device, and, at a later time, writing this data at electronic speeds into the main memory. It must also be capable of reading in data at electronic speeds from the main memory and writing this data slowly at the output device.

Tape or Punched Card Handling Equipment

As might be expected, tape or card handling equipment varies from manufacturer to manufacturer. Certain general characteristics, however, remain true for the equipments in each

category. These general characteristics are given primary consideration in this chapter.

PUNCHES.—Tape or card, punches are actuated by a manual keyboard, or in some cases by the computer. The keyboard buttons, when operated, open and close groups of switches which control solenoid-operated punches (figure 7-20). Once the holes have been punched in one section, the punch automatically advances the card so that the next section is located under the punching station punches.

A representative card punch is shown in figure 7-21. Punches have a check read station to verify the data punched. Each has a buffer register (not shown) to receive data from the computer. The computer will place the data it wishes to send on the data lines and set a signal on another line to notify the punch unit. The punch unit samples the data from the computer and sets a signal line informing the computer of its action. In addition some form of interrupt communication will be used to keep the computer informed of the functional status of the punch i.e., op/inop, broken tape, card jam, punching error, etc.

With some machines, skipping, duplicating, and shifting can be performed automatically by the program unit of the punch, thus obviating

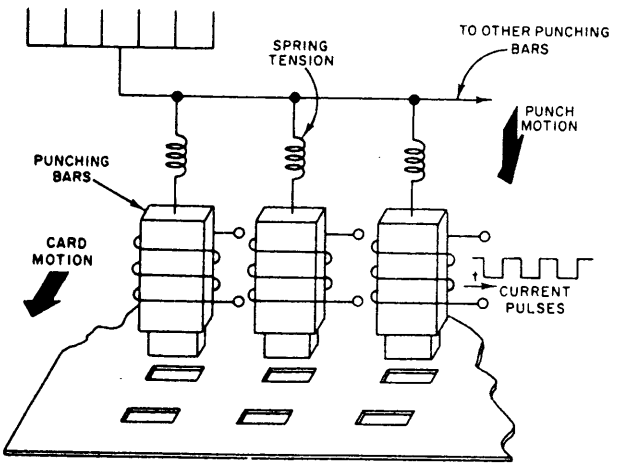


Figure 7-20.—Card Punching Mechanisms.

the necessity for including these functions in the computer program. Each of these operations is controlled by a coded prepunched program card or tape. Punched holes in certain zones and digits in the prepunched program cause certain sections to be skipped on the card or tape being punched. The prepunched program also causes automatic duplication of material. Note that the use of a prepunched program in this manner reduces the manual effort required during repetitive punching.

READERS.—Tape or card readers are used to translate the holes in a punched tape or card back into the signals that represent characters. A reader is sometimes used as a computer input device, although it is more frequently used to operate an electric typewriter or a printer in order to translate punched data back into written form. Readers are also used in conjunction with conversion devices which convert and transfer data onto some other medium.

One method of card reading consists of passing a punched card between a contact roller and spring-mounted electrical contacting fingers or brushes. When a punched hole in the card passes the reading station, the brushes make contact with the roller underneath, as shown in figure 7-22. The cards move past a column of brushes (often called a reading station). A circuit is completed (closed) each time a brush is permitted to pass through the hole and make contact with the roller. Thus, the circuit closes when a hole is present, and opens when no hole is present. Card equipment devices use this open-close indication to actuate other control and data-processing circuitry. There are as many brushes as there are code areas in a single column on the card being read.

The techniques used with punched tapes are almost identical to those used with punched cards. The holes that are punched in the paper tape however, are round rather than rectangular in shape.

Punched tapes (figure 7-23) are often used with military and commericial data-processing systems. The advantage of this form of data

49.5.2X

Figure 7-21.—IBM Type 29 Card Punch.

recording is that most business machines can easily be modified or designed to punch tape records of their operations. These tapes are punched in a modified teletype code that can be transmitted over wire or radio teletype lines.

Many large installations use punched tape units as preliminary data-processing devices at the point of transaction. All pertinent data are sent by teletype to a central data-processing installation. The punched tape data can be converted and read onto magnetic tape (or some other suitable medium), then fed into a large computer.

Tapes are available in several widths (7/8 ", 1 ", 1-1/8 ", etc) and in 1ØØ-foot, 35Ø-foot, and 7ØØ-foot rolls. A sprocket channel (a line of small circular holes that are punched on the tape at the time data is recorded), appears longitudinally along the length of the tape, as shown in figure 7-23. These holes are engaged by sprockets that drive the tape past punching and reading stations. Reels are used to store the tape.

The speeds used with paper-tape equipment are, in general, lower than those used with punched cards. A high-speed paper-tape punch

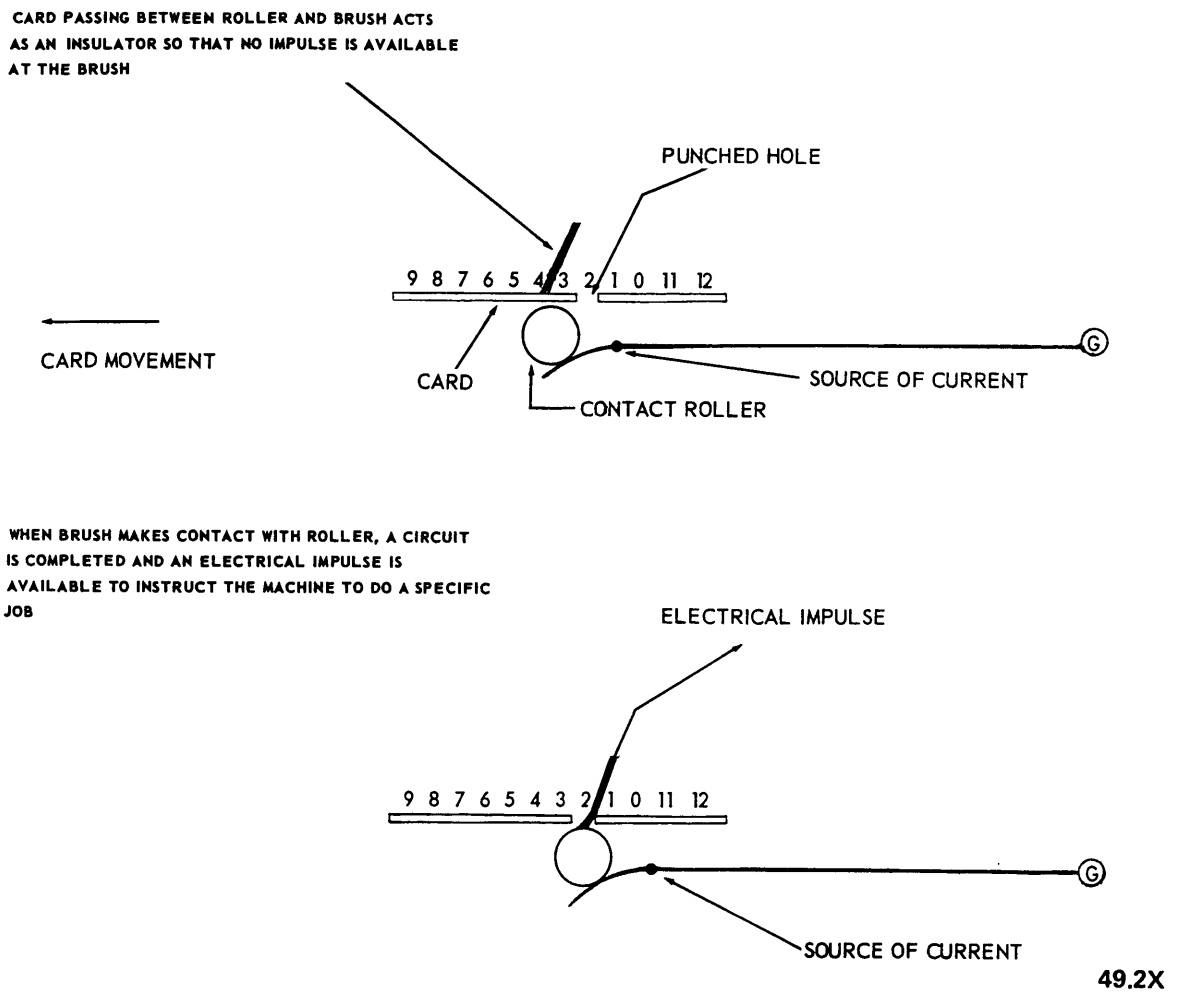


Figure 7-22.—Brush Reading Punch in Row 2.

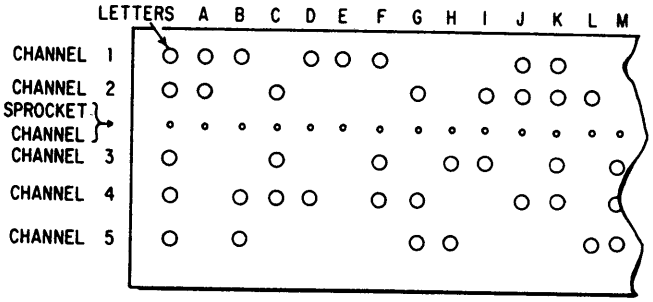
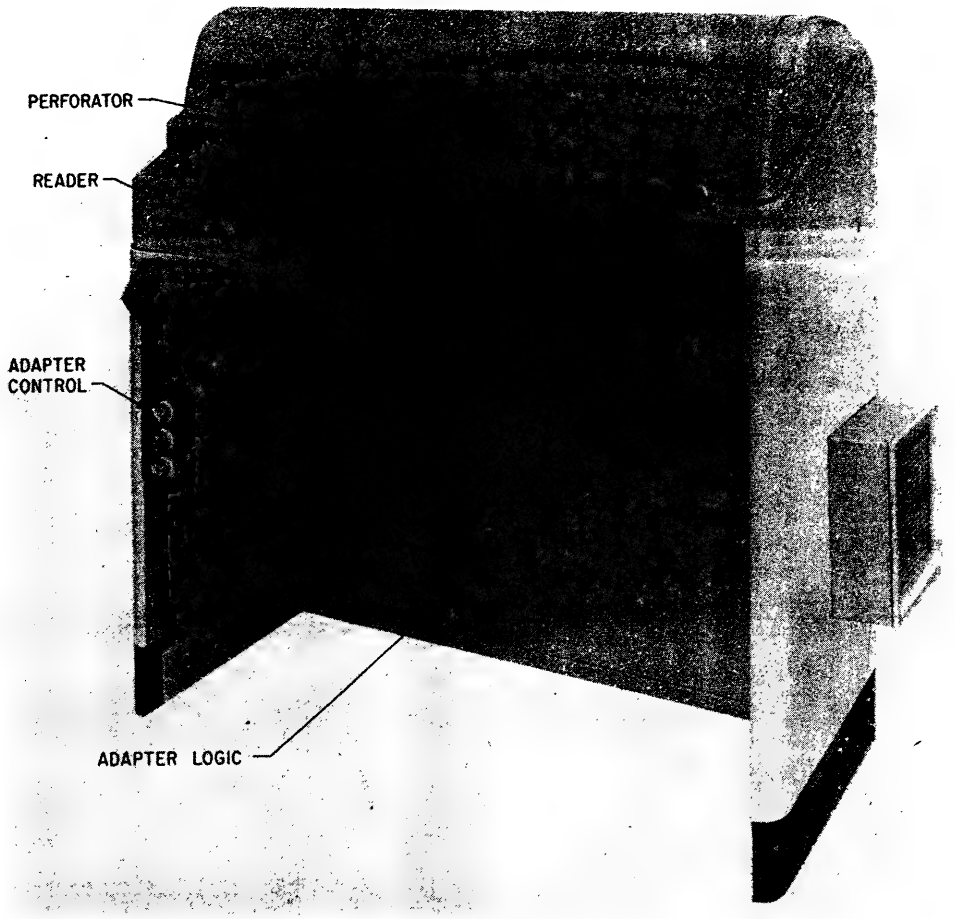


Figure 7-23.—Five-Channel Paper Tape Code.

used with some commercial computers can punch at a rate of 12Ø characters per second. Most mechanical punches punch from 1Ø to 6Ø characters per second.

A type of reader which uses the photoelectric principle (rather than mechanical) provides the fastest means of reading punched tapes. With this method, a light-sensitive material is placed beneath each of the longitudinal channels and the sprocket channel (figure 7-21). A light placed above the tape and over the light-sensitive material causes an output signal to be produced from each channel in which a hole has been punched. The signals are



1.217.6

Figure 7-24.—Teletypewriter Set AN/UGC-6, with Adapter.

amplified and fed to the computer as input information.

The sprocket hole outputs (which occur at each reading point on the tape) signal the read interval. Therefore, each channel output is sensed when a sprocket hole passes the station. The tape moves continuously until it is ordered to stop by a STOP instruction.

Photoelectric type readers can feed up to 1ØØØ characters per second into the computer. This speed is comparable with that of the fastest card reader available.

Tape or card readers for on-line operation will usually have a buffer storage register to permit packing the data read from the tape or card and sending it to the computer in a full word length (unit). Communications with the computer are on a data ready basis. (The information is packed and placed in the output registers of the I/O device after which a status line is "set." The computer reads the data and places a signal on another line indicating that it has received the information and the reader can continue.) There may be provisions for the reader to send functional status reports to the computer such as operative/inoperative signals, card jam, broken tape, etc. This is generally accomplished by placing a specific code in the output register and setting an interrupt line. This action informs the computer regarding the status of the I/O device.

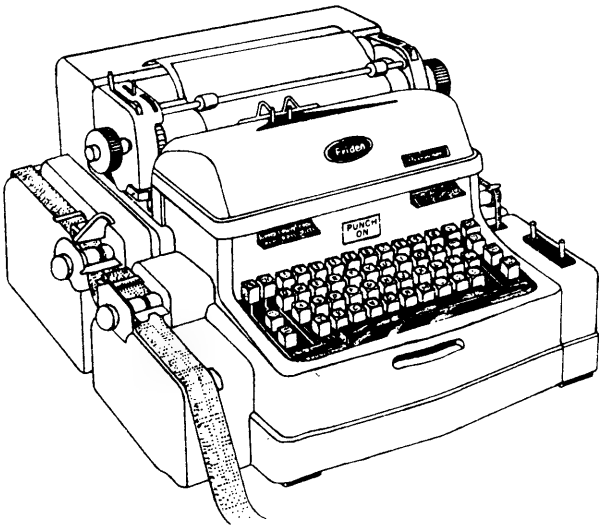
Keyboard Inputs

Various types of keyboards have been either adopted or designed for on-line computer use. Since they are operated by humans, they are characterized by an extremely low data rate. The use is therefore generally limited (in on-line input applications) to insertion of irregularly occurring parameters and commands. For large amounts of data, they are used off-line to prepare a tape which can be used to feed the information into the computer.

TELETYPE.—Teletype equipment using either the standard 5-level or the ASCII (American Standard Code for Information Interchange) 8-level code, covered in Chapter 5, may be utilized as I/O devices. An adapter unit is installed in the teletype unit at a computer site to convert each teletype character into parallel format and supply various signals relating to the status of the equipment. Such a machine is shown in figure 7-24. When the code is converted to parallel format, the start and stop characters are dropped and only the five intelligence bits are transmitted. For output, the machine provides a printed copy and a punched tape.

FLEXOWRITER.—The flexowriter is another device similar to the teletype, figure 7-25. Unlike the teletype, the flexowriter communicates using a six bit parallel code. It is normally used as an off-line device for paper tape preparation and correction. The flexowriter also communicates at 10Ø words per minute and has provisions for either on-line or off-line preparation of tape while typing. On-line usage requires an adaptor unit for computer communication.

MISCELLANEOUS KEYBOARDS.—Various special I/O keyboard devices are used. Generally they are used to communicate with a running program. They may be used only as input devices (equipped to provide some indication that the data has been accepted or rejected), or they may have some provision for readout of data from the computer. The keyboard and character keys may vary from device to device.



164.47

Figure 7-25.—Flexowriter.

High Speed Printing

LINE-A-TIME PRINTERS.—As the name implies, the line-a-time printer is a machine that prints an entire line at one time. In general, these machines can print a page faster than a typewriter because the operation of printing each character on a line is performed simultaneously rather than separately.

There are two major types of line-a-time printers: type bar printers and type wheel printers.

Type Bar Printers.—A computer-operated electric typewriter can print at speeds of approximately 10 letters or numbers (referred to as printed characters or digits) per second. Even though this results in printing 6ØØ characters per minute, as compared with the 3ØØ characters per minute that are produced by the average human typist, the electric typewriter is not fast enough for all operations.

The bar printer (figure 7-26) is a variation of the typewriter. The type characters protrude from the edge of a bar. A character is printed by a hammer which strikes the paper and presses it against an inked ribbon and the type bar.

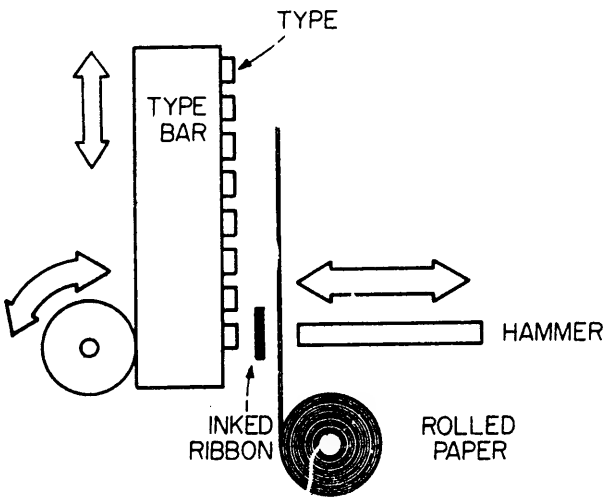


Figure 7-26.—Bar Printer.

The type characters are mounted so that one appears above the other. A particular character is selected for printing by raising and lowering the type bar so that the desired character is lined up with the hammer.

Groups, or "gangs," of type bars are used to make up a multibar printer. The number of type bars that are used depends upon the number of characters that must be printed simultaneously. The bars are geared so that they are similar to toothed racks. The digitally operated pinions (not shown) that move the racks are driven by two stepping-switch mechanisms (one for each direction of motion).

Let us assume, for example, that the letter S is printed by stepping the type bar ten times. An S in computer code must be converted into ten stepping pulses which advance the type bar ten times. The printing hammer is then actuated and the letter is printed. After printing, the type bar is returned to a suitable rest position, or it is permitted to remain at the S position until the next stepping pulses are applied. These new stepping pulses are automatically corrected to allow for the fact that the bar is presently at the S position.

Type Wheel Printer.—Another type of printer has the character type on the edge of a wheel (figure 7-27). As many wheels as desired are arranged in parallel. The type wheels are continuously rotated. The striking of the hammers against the paper and ribbon is synchronized with the wheel rotation so that certain characters are selected for printing. This arrangement decreases the time required to position characters under the hammers, and provides higher printing speeds than the gang printer.

Photographic Printing

Photographic printing is accomplished by photographing the display produced by any one of several types of electronic character-writing tubes. The basic principle is illustrated in figure 7-28A. The negatives produced using this process can be used to make photographic prints or for offset printing.

The electronic character-writing tube and its associated circuitry (not shown) receive binary signals that represent alphanumeric characters. Translating circuits operate a cathode-ray (display) tube that is designed to display the alphanumeric characters in English letters and Arabic numbers.

The simplest display tubes use a form of grid or matrix (figure 7-28B) which, in effect, is an electronic stencil. The electron beam is aimed at a specific character in the metallic grid plate. The holes in the grid are in the shape of the characters to be printed. When the beam is

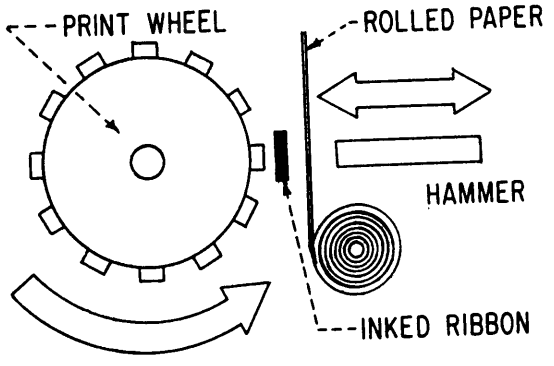
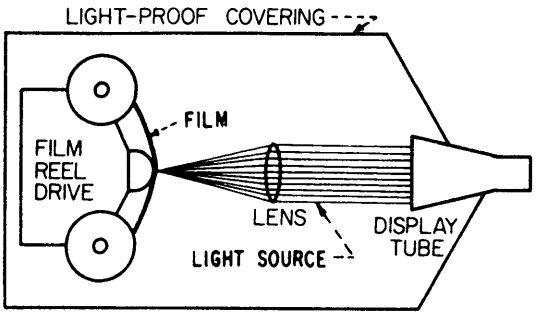


Figure 7-27.—The Type Wheel Printer.



A BASIC PRINCIPLE

B MATRIX (GRID)

124.88

Figure 7-28.—Photographic Printer.

deflected to the desired character the electron beam that passes through the hole has the shape of the character. A second deflection positions the character-shaped beam to its proper position on a flourescent screen. This process is continually repeated, so that the screen displays a series of characters which are photographed.

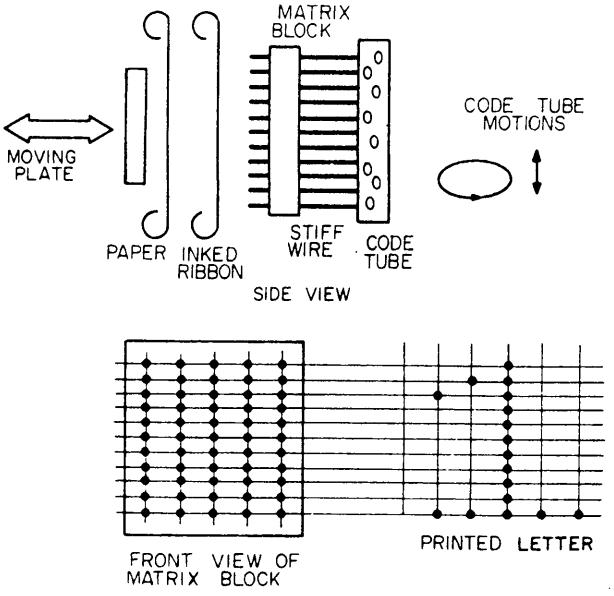
Other photographic systems use waveshaping circuits which generate voltages to approximate the component lines of each character, so that a simple cathode-ray tube can be used to display characters.

Both of the cathode-ray tube arrangements that have been discussed require elaborate beam-gating and deflection systems. However, they possess the advantages of being extremely fast and of requiring no mechanical moving parts.

The camera is usually contained in a light-proof box that surrounds the face of the tube. An optical system is used for recording the tube display on film. The film is reel-fed, and is stepped to the next frame each time that the tube message changes. A shutter is not required, since the cathode-ray tube blanking pulses perform the same function. The film-drive mechanism is the only mechanical device in this entire system.

Wire Punch Printer

The wire punch printer uses a matrix of stiff wires to print letters or to punch holes (figure 7-29). By using the proper combination of wires, it is possible to punch any character or letter. The wires of each matrix are bundled together and are run to a cylindrical code tube that has holes machined into its surface. The wires are fanned out along the longitudinal axis of the



124.89

Figure 7-29.—Wire Punch Printer.

tube, with each wire resting against the surface of the cylinder. When a particular character is to be printed, the tube is rotated and moved along its axis so that the correct combination of individual wires is backed by metal, and the other wires are located over holes. At the printing end of the wire bundle, a moving plate pushes paper into contact with the matrix of wire ends. Those wires that are backed by metal will exert sufficient force on the paper to print the character. The remaining, unbacked wires will not exert this force.

If the plate is smooth, and an inked ribbon or carbon paper is placed between the wires and the paper, the backed wires will press against the ribbon and will print the appropriate character as a group of black dots. The wires are spring loaded (not shown on the diagram) so that they will return to their original positions when the plate pressure is removed.

BASIC COMPUTER OPERATION

With an understanding of the function of the various computer component units, let us now consider a basic computer instruction and how this instruction is executed. Let the instruction be as follows:

> "Add the contents of the A-register to the contents of memory address location 123 and store the results in address 456 in memory."

We will assume that the computer used is the stored program type and that all instructions, data, numbers, and symbols have been previously loaded or stored in memory at known addresses. The stored input may have been read from a magnetic tape (similar to that used with commercial tape recorders), from punched tape or cards, or other long-term storage.

If the instruction to be executed is the first programmed operation, energizing the start button will cause the control unit to issue an order "Read Instruction." The instruction will be read into a register in the control unit where it will remain throughout the execution cycle.

Note that the mathematical operation requested in the instruction is ADD. The instruction word thus contains a code which is interpreted by the control unit as ADD.

After reading the instruction, the control unit automatically energizes circuits which will (1) read-out the contents of memory address 123, (2) transfer this information to a register (say the X-register) in the arithmetic unit, and (3) perform an add X to A operation. The ADD process is thus accomplished, being constantly monitored by the control unit to ensure that no further actions are initiated before the ADD operation is completed. The results of the ADD operation are stored in the accumulator from which, by control request, it is transferred to address 456 in memory. This ends the instruction. The control unit will read and execute the next instruction.

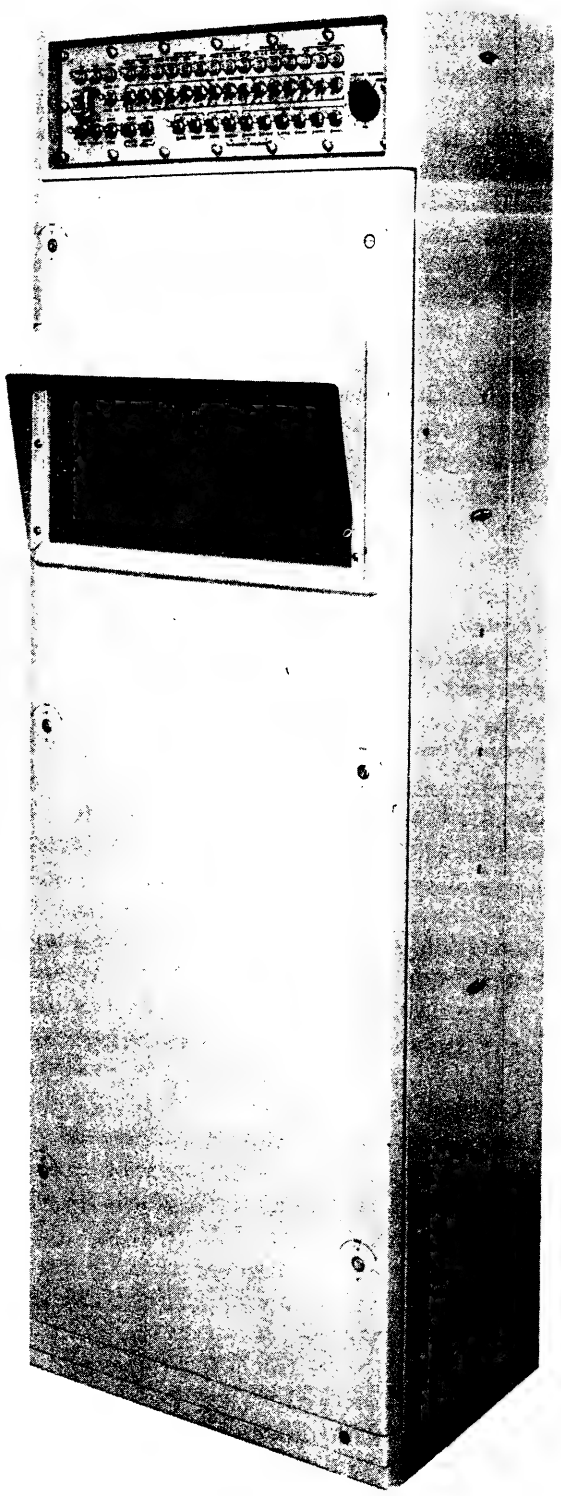
If the result is to be displayed at the output immediately or at a later time (as stipulated in the programmed instructions) the control unit upon receipt of the instruction will issue an order to read-out the contents of memory address 456. Because read-out (which sometimes involve printing by some electromechanical apparatus) is extremely slow as compared to computer speed, most computers use a secondary storage device called a buffer into which data is read directly from the primary (main) storage at computer speeds. When read-out is desired, the control unit enables the buffer storage to read-out all or any part of the buffer storage data. The buffer read-out is independent of the main computer operation, and in some computers only one instruction is required to start and stop the read-out process.

DIGITAL COMPUTERS WITHIN THE NAVAL SECURITY GROUP

There are many different types of digital computers in use within the Naval Security Group. The following text discusses two of these and is presented here only as representative examples of the types which are used.

DIGITAL DATA COMPUTERS CP-771(V)/UYK-3(V) AND CP-771A(V)/UYK-3(V)

This section describes the purpose, capabilities, and physical characteristics of Digital Data Computers CP-771(V)/UYK-3(V) and CP-771A(V)/UYK-3(V) (figure 7-30) manfactured as the Model 133B and Model



1.456

Figure 7-30.–Digital Computer, CP-771/UYK-3A(V).

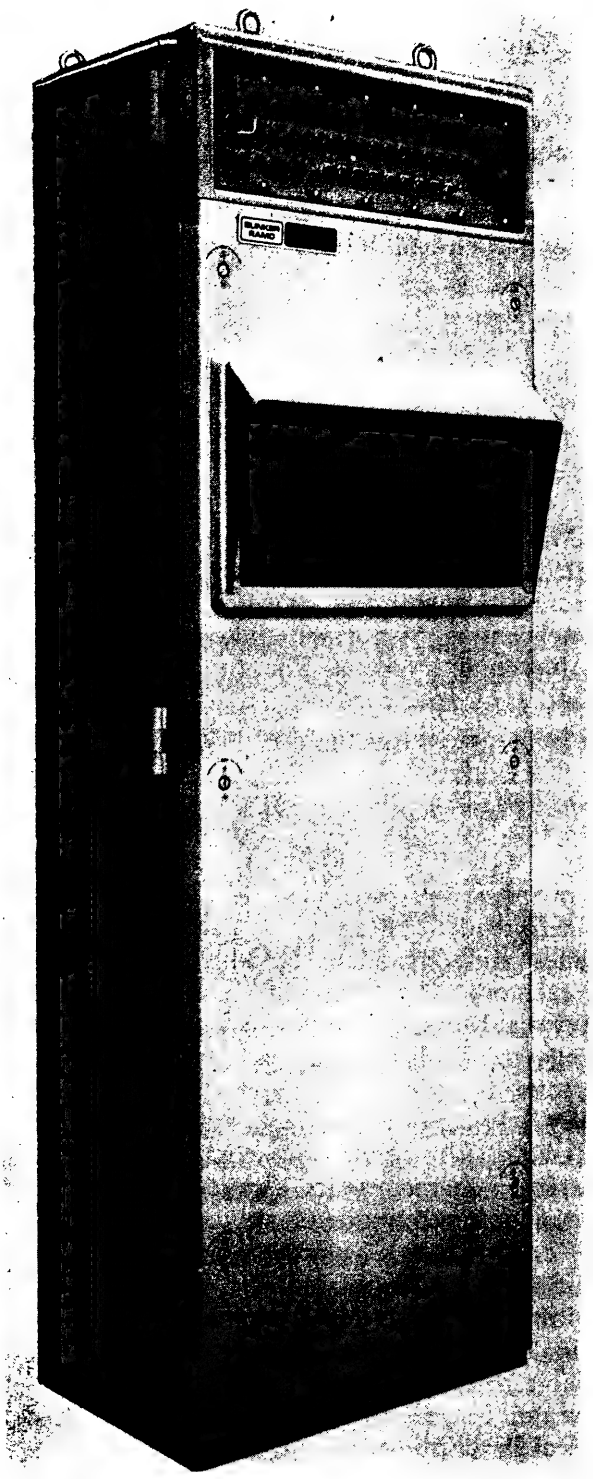
133C Digital Computers, respectively. Both models of the computer are similar in design and function; as a result no distinction has been made between the two models and it will be referred to simply as the CP-771(V)/UYK-3(V), except where differences exist. In those cases where the models differ, the computers will be referenced by either 133B or 133C model number.

Purpose and Capabilities

The Digital Data Computers CP-771 (V)/ UYK-3(V) and CP-771A(V)/UYK-3(V) are stored-logic, multiple-purpose digital computers that meet the military standards of quality and reliability and conform to Naval Tactical Data System (NTDS) data-processing requirements. The computer utilizes a random-access, 16,384-word magnetic core memory and six 15-bit flip-flop registers for internal data processing. Internal operations of the computer is synchronous with clock pulses occurring at 1-microsecond intervals. The time required to read-store or to clear-store is 2 microseconds, with most logical commands (logands) executed in 4 or 6 microseconds. In addition, the Model 133C has the added capability of being connected to the Buffer-Extended Memory Unit, MU-591/UYK-3(V), (figure 7-31), commonly referred to as the BR-174 Buffer-Extended Memory or just BEM which provides an additional 16,384-word magnetic core memory. The basic word length is 15 bits with variable word lengths in multiples of 15 bits up to 45 bits. The computer command repertoire consists of approximately 8500 logands. Except in a few instances where germanium semiconductor components have superior characteristics, solid-state silicon semiconductor components are used throughout the computer. Special cooling or air conditioning is not required due to the low-power consumption. The computer operates within an ambient temperature range of Ø to 5Ø°C.

Physical Description

The computer is contained in a drip-proof, cast-aluminum cabinet consisting of three



1.457

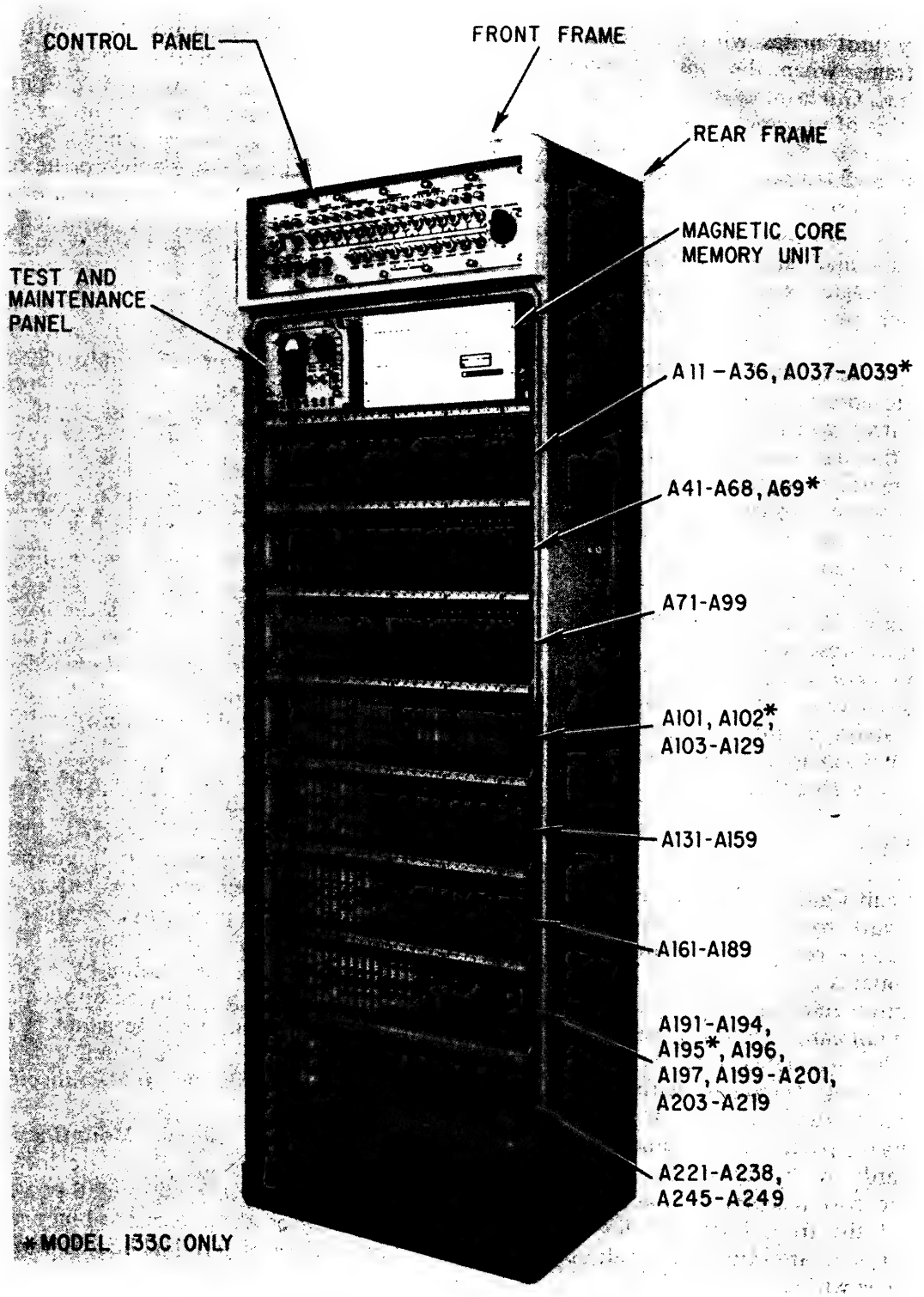
Figure 7-31.—Digital Computer, Buffer Extended Memory, MU-591 Used With the CP-771/UYK-3A(V).

assemblies: the hinged front frame, the rear frame, the access door. Modular construction is used throughout the computer with all electrical circuits contained on circuit cards or replaceable assemblies. Four mounting holes in the base of the cabinet and two at the rear are provided for securing the computer in its mounting position. The mounting, the rigid, cast-aluminum cabinet, and the method of securing assemblies within the cabinet provide maximum protection against shock and vibration. Four removable screw eyes at the top of the cabinet are provided for lifting, moving, and installing the computer. Access to the computer circuits is from the front of the cabinet and all inter-connecting and power cable connectors are located at the rear. Cooling is provided by an internal blower, with air intake from the front and exhaust to the rear.

FRONT FRAME.—The Front Frame (figure 7-32) contains the control panel, magnetic core memory unit, memory unit current source, test and maintenance panel, and eight rows of circuit cards. The front frame is hinged on the left of the rear frame and is secured by five captive screws. Four of these screws are located along the right side of the front frame and one is located at the bottom. When the screws are loosened, the front frame can be opened to provide access to the back of the front frame and to the assemblies located on the rear frame. A door-stop arm, connected at the top between the two frames, limits the distance that the front frame can swing outward. The doorstop arm also holds the front frame in the open position.

Control Panel.—The control panel contains all switches and indicators required for computer operation. The control panel is located above the access door at the top of the front frame where it is accessible and visible at all times. Sixteen computer circuits by cable connectors permits easy removal of the control panel from the cabinet.

Magnetic Core Memory.—The magnetic core memory unit is a completely enclosed assembly containing four 16-plane core matrices, two heating elements, three thermocouples, and a temperature monitoring circuit. The memory unit is secured to the right hand side of the front frame under the control panel with two



1.458

Figure 7-32.—Front Frame of Computer With Access Door Removed.

thumbscrews and two hand jackscrews. An input/output connector at the base of the memory unit mates with a receptacle on the front frame when the unit is secured in its mounting. Guide pins ensure proper alignment and mating of the connector.

Test and Maintenance Panel.—The test and maintenance panel contains the controls, indicators, and test point jacks required for testing and maintaining the computer. On Model 133C a toggle switch is provided to allow the computer to operate normally when not connected to the BEM (Buffer-Extended Memory). The panel is located to the left of magnetic core memory unit immediately below the control panel. Four captive thumbscrews secure the test and maintenance panel to the front frame. Computer interconnection to the panel is made through a cable connector, which allows the test and maintenance panel to be easily removed.

Memory Current.—Memory current source comprises those components that are required to develop a constant, magnetic-core drive current. This assembly is mounted inside the front frame, immediately behind and above the control panel. Writing to the memory current source is made by a single cable connector, making the current source a modular and replaceable assembly.

Circuit Cards.—Circuit cards are mounted in eight card rows located below the test and maintenance panel in the front frame. Each card row contains 29 receptacles and associated guide slots into which the circuit cards are inserted. The circuit cards are firmly seated in or removed from the associated receptables by means of special tools supplied with the computer. Retainer plates are used to hold the cards securely in place. Each retainer plate is secured to a card row by five captive screws. Wiring to the card row receptacles is accessible from the back of the front frame. Solderless, wire-wrap connections are used for all circuit care receptacle wiring.

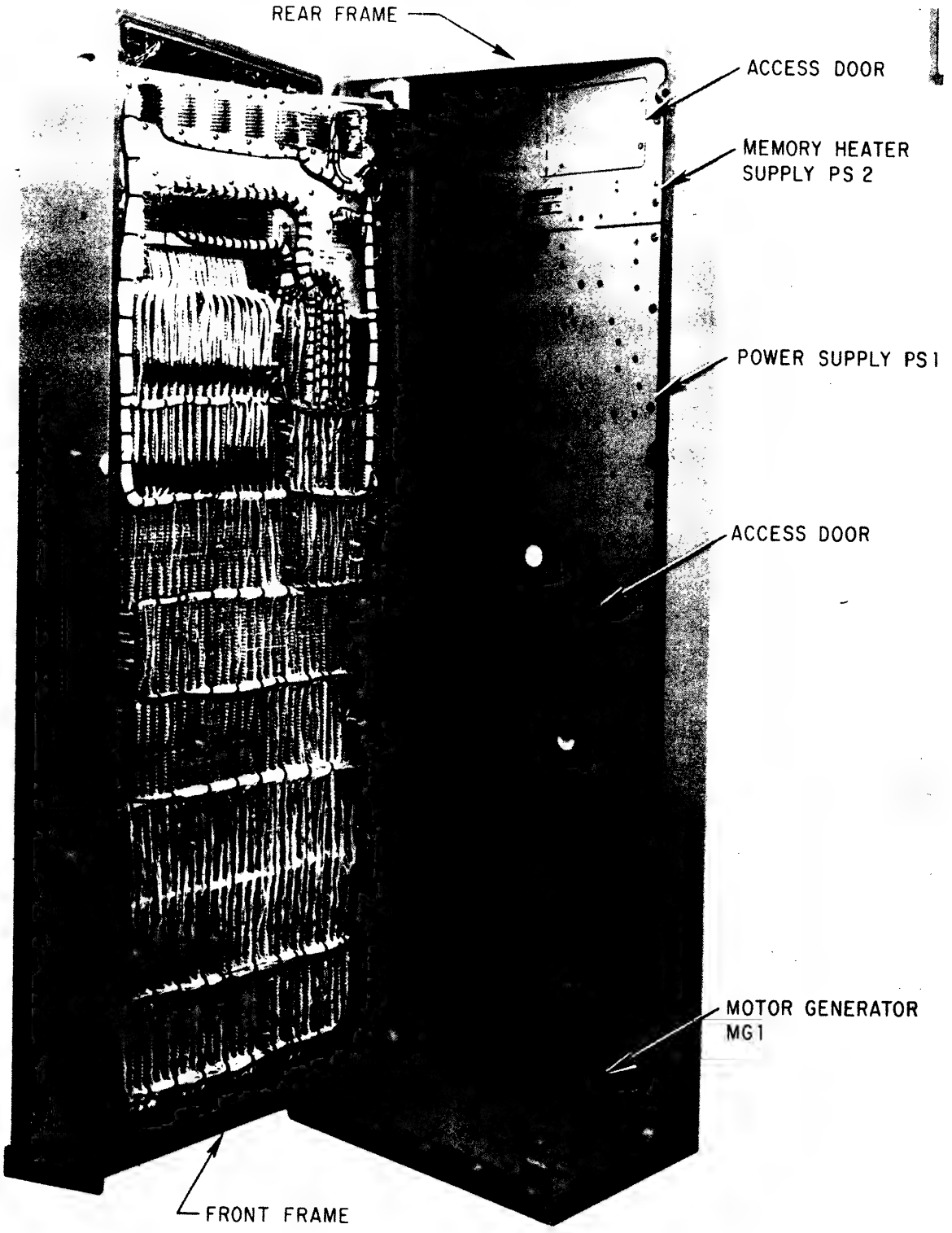
REAR FRAME.—The rear frame (figures 7-33 and 7-34), which is the main structure of the cabinet, contains two power supply assemblies, a motor-generator, a power transfer panel, two terminal board mounting plates, an input/output connector panel (in the Model 133C, there are two input/output connector panels, designated as upper and lower input/output connector panels), and an auxiliary connector plate. Two holes in the rear and four in the base of the rear frame are provided for installation purposes. A screened air-exhaust port is also located on the back near the top of the rear frame.

Memory Heater Power Supply.—The memory heater power supply PS2 contains the power supply and control circuits that supply heater power to the memory heater. This power supply, located at the top of the rear frame, consists of a panel assembly upon which are mounted the power supply components. The control circuit is contained on a circuit card which is also mounted on the panel assembly. The panel assembly is mounted so the component side is towards the back of the rear frame and secured in place by six captive screws. A hinged access door is provided on the panel assembly for access to the circuit card, the power supply cable connector and fuse.

Power Supply PS1.—Power supply PS1 produces d.c. power for the computer circuits. This power supply, located below memory heater power supply PS2, consists of power supply components, regulation control and power supply protection circuit cards mounted on a panel assembly. The panel assembly is secured to the rear frame by 12 captive screws with the power supply components mounted on the inside surface of the panel. A hinged access door, located on the panel, provides access to the circuit cards and cable connectors.

Terminal Board Mounting Plates.—Two terminal mounting plates are located on either side of the rear frame behind power supply PS1. Routing of the internal cabinet wiring is made by the use of terminal boards mounted on these plates.

Auxiliary Connector Plate.—An auxiliary connector plate is provided at the back of the



1.459

Figure 7-33.—Rear Frame of Computer with Access Cover in Place.

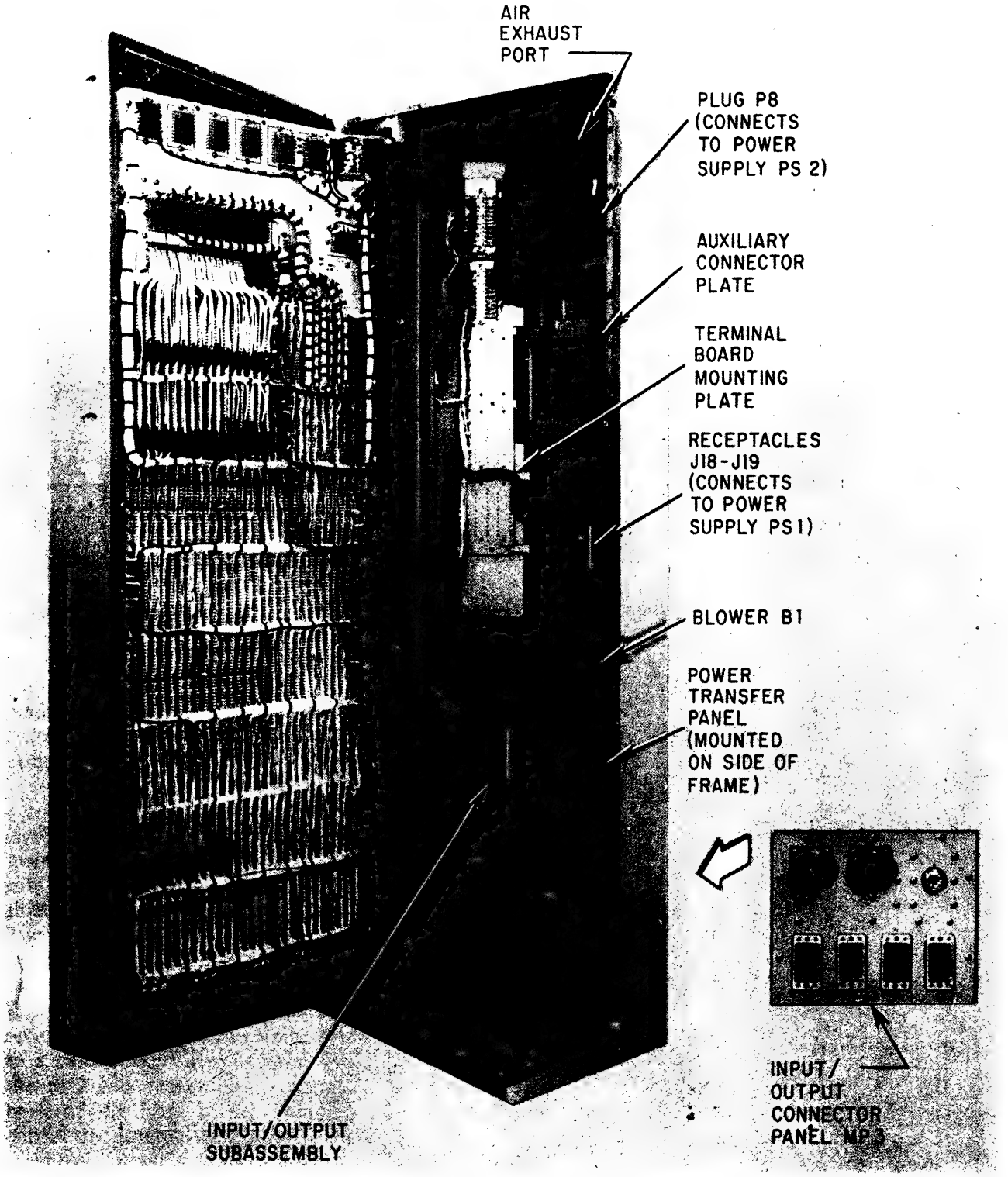


Figure 7-34.—Rear Frame of Computer Showing Location of Components.

rear frame behind power supply PS1 (Model 133B only). This connector plate can be removed and replaced by a connector panel if additional cable connection facilities are required for the computer.

Blower B1.–Blower B1, located immediately below power supply PS1, produces the cooling air flow through the cabinet assembly. This unit is a tandem blower operating from the primary ac input power. Access to the blower is gained by removing the panel located between the power supply PS1 and motor generator MG1.

Power Transfer Panel.–Power transfer panel, located on the right side of the rear frame immediately below blower B1, contains power control relays K1 and K2 and two terminal boards. Distribution of primary ac power, under control of the control panel power application switches, is made through the contacts of relays K1 and K2 and the terminal board connections. Access to the power transfer panel components is accomplished by removing the blank panel located between the power supply PS1 and motor generator MG1.

Input/Output Connector Panel.–Input/output connector panel MP3 (referred to as the lower input/output connector panel in the Model 133C), located on the back of the rear frame, contains the connectors required to interconnect all input and output cables and the primary power to the computer.

Upper Input/Output Connector Panel.–The upper input/output connector panel, located on the back of the rear frame, contains the connectors required to interconnect the Model 133C with the BEM (Buffer-Extended Memory).

Motor Generator MG1.–Motor generator MG1, located in the base of the rear frame, regulates primary input power applied to power supply PS1. This motor generator converts the input power to a 3Ø-volt, 41Ø-hertz, 3-phase output which is then applied directly to power supply PS1. The motor generator is secured to the base and sides of the rear frame by means of shock mounts.

ACCESS DOOR.–The access door (figure 7-35), covers all of the computer front frame below the control panel. When the access door, which is secured to the front frame with six pawl fasteners, is removed, the maintenance panel and circuit cards are accessible. An air-intake port that permits the input of cooling air into the cabinet assembly is located at the top of the access door. This port is covered by an air filter to reduce the entry of dust and foreign particles into the cabinet assembly, and a radio-frequency interference filter to reduce the transmission of RF interference through the port. Inside the access door are provisions for the mounting and storage of spare lamps, fuses, and special hand tools required for normal maintenance of the computer. The special tools that are provided are those required for opening the front frame, and for removing the retainer bars and circuit cards.

CP-771(V)/UYK-3(V)
Functional Description

This computer is controlled in operation by stored logic. In the stored logic technique, much of the control logic that is normally wired into the machine is, instead, stored in memory as part of the program. Only those common logical sequences required to maintain program control are wired into the machine as implicit micro operations.

Control logic is stored in memory in the form of logands (logical commands). A logand is a 15-bit word that is coded to activate various combinations of implicit micro operations. A series of logands can, therefore, be stored in memory together with operands or data words to carry out a particular computer function. In this manner, the program stored in memory determines what operations the computer performs to solve the particular data processing problem at hand.

Logands occupy a single location in memory, as do all 15-bit operands (data words) or address words. Logands, with their operands and address words, are normally stored in sequential memory locations. The logand is read from memory, interpreted, and the specified micro operations are executed.

COMPUTER TIMING.–Figure 7-36 is a functional block diagram of the

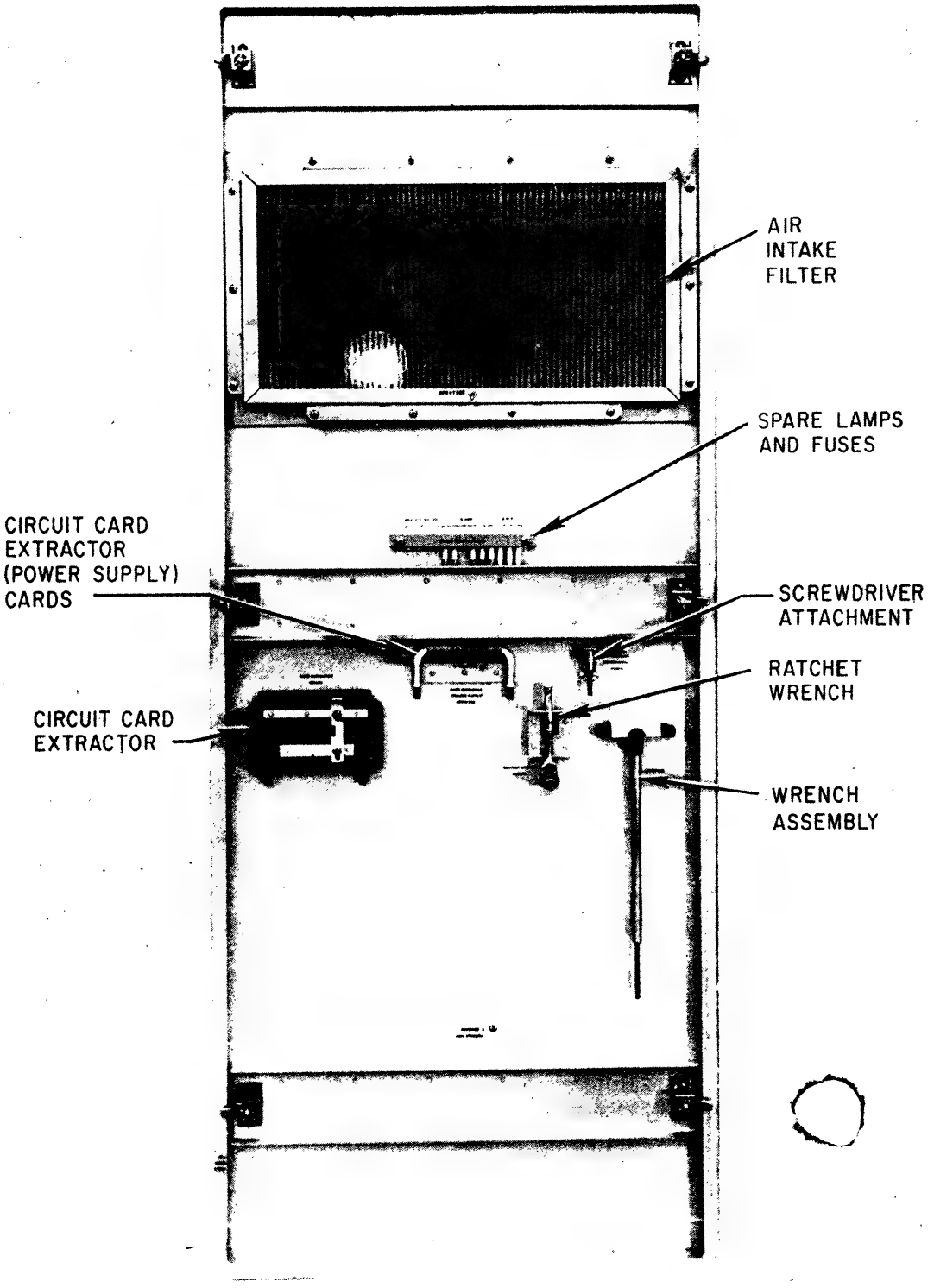
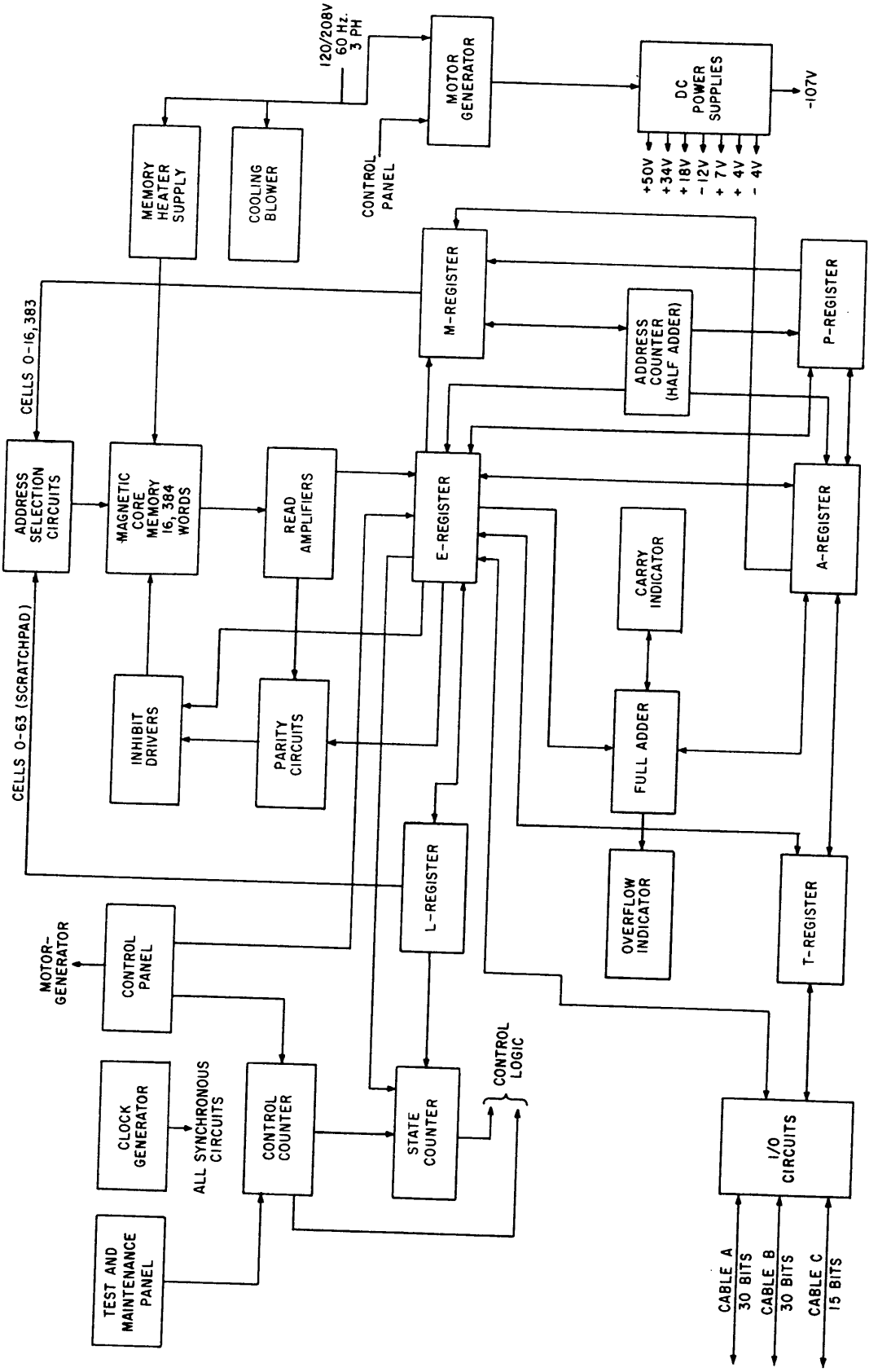


Figure 7-35.—Computer Front Frame Access Door.



1.462

Figure 7-36.–Functional Block Diagram of the Digital Data Computer CP-771(V)/UYK-3(V).

1.461

CP-771(V)/UYK-3(V) which is a synchronous machine, deriving its primary timing from a 1-megahertz clock pulse. Micro operations are sequenced by either the control counter or the state counter.

Computer Clock.—The clock pulses are generated by a crystal-controlled oscillator operating at a 1-megahertz frequency. Clock pulses normally occur at 1-microsecond intervals, although the normal clock output can be inhibited by control logic thereby reducing the clock frequency or stopping the clock altogether.

Control Counter.—The control counter sequences computer micro operations which are selected by switches on the control or maintenance panels during the manual mode. The control counter also provides for an orderly entry into, and exit from, the running mode. During the running mode, computer operation is under complete control of the program being executed.

State Counter.—The state counter sequences computer micro operations which are selected by the logand being executed during the running mode. A logand is executed during a state counter cycle. The type of logand and the method of addressing (direct or indirect) determines the number and sequence of states in a state counter cycle. The micro operations of the logand involved are executed during specific states of the resulting state counter cycle.

COMPUTING ELEMENTS.—Internal computing elements comprise those circuits which are used for storing, handling, and acting upon logands and data within the computer. The principal computing elements include the magnetic core memory, six 15-bit registers, a 15-bit parallel full adder with carry and overflow indicators, and a 15-bit parallel half adder which serves as an address counter.

Basically, the computer performs the computing operations, including arithmetic functions, by transferring and shifting data in parallel between registers. The register transfer and shift operations are selected by the logand being executed, and sequenced by the state counter cycle for that logand.

Magnetic Core Memory.—The memory provides random access storage for 16,384 computer words. The memory is a destructive read out (DRO) memory and, therefore, requires that each read operation be followed by a write operation to restore the contents of the accessed memory location. Read and write operations are synchronized to the clock, and each operation requires 1 microsecond to be completed. Therefore, a complete memory cycle requires 2 microseconds.

Computer words are stored in memory with a 16th bit added for a parity check. During each write operation, the status of the 16th bit is established such that an odd number of 1's are contained in the 16-bit word stored in memory. During each read operation, the 16-bit word is examined to determine whether the number of 1's is odd or even. An even number of 1's sets a parity error indicator.

M-Register: The M-(memory address) register holds the address of both logands and operands. The contents of the M-register are decoded by address selection circuits to select any one of the 16,384 memory locations. Nonsequential memory addresses are transferred into the M-register from either the E-, P-, or A-registers. Sequential memory addresses are transferred into the M-register from the address counter.

Address Counter: The address counter is a 15-bit parallel half adder. The contents of the M-register are always inputting to the half adder, while a logically controlled carry input causes either a 1 or Ø to be added to the present address. Sequential accessing requires that a 1 be added to the contents of the M-register, and the sum is transferred to the M-register at the start of the next memory cycle. When a non-sequential address is transferred to the M-register from another register, the incremental contents of the M-register are usually saved by transferring them to either the P- or A- register.

E-Register: The E- (memory exchange) register is the only register through which

computer words are read from or written into memory. Therefore, micro operations are provided to exchange the contents of the E-register with any of the five remaining registers. The E-register also serves as the input buffer, with the T-register handling the least-significant half of the 30-bit input words. The logand is partially decoded in the E-register while the logand is being written back into memory. The E-register also holds one of the operands used in arithmetic, merge, and extract logands.

L-Register: The L- (logand) register is used to hold logands while the logands are being interpreted and executed. Inputs to the L-register are exclusively from the E-register. Because the coding of the logand determines the sequence of computer operation, the L-register outputs are used extensively. The six most significant bits (10 to 15) of the logand in the L-register contain the code for the primary command which determines the basic sequence of micro operations involved in the execution of the logand. Bits 7 to 9 of the logand in the L-register contain an address option which determines the memory address that is used in accessing memory for the operand. Bits 1 to 6 of the logand in the L-register can contain a memory address (cells 0 to 63). These cells are referred to as "scratchpad" because the cells provide easily accessible, nonsequential storage for intermediate results, control address, and other miscellaneous data. Bits 1 to 6 of the logand in the L-register can also contain memory or shift control bits, a shift count (the number of places to be shifted), a condition code, or a secondary command.

A-Register: The A- (arithmetic) register is used to hold the sum in the addition operation the most-significant part of the product in the multiply operation, the remainder after a division operation, and the result of some logical operations. During arithmetic operations, the A-register receives data from the full adder. The A-register can also receive inputs from the E-, P-, and T-registers, or the address counter. The A-register can be shifted left or right, either alone or with the P-register. The A-register can also be used as a temporary storage location for memory addresses.

Full Adder: The full adder is a 16-bit parallel adder which forms the sum of the contents of the A-, and E-registers and the carry indicator. The full adder performs this addition at all times; however, the full adder output is transferred to the A-register only when specified by the program. The 16th bit of the full adder is normally not transferred to the A-register, but is used in making comparisons of operands in the A- and E-registers.

Carry Indicator: The carry indicator holds the carry input for the first full adder bit position, and also stores the carry from the 15th bit position resulting from the addition operation. Addition operations for 15-bit words, or for the least significant 15 bits of multiple-length words, reset the carry indicator to 0 before the addition process occurs. Addition of the intermediate or most-significant 15-bit portion of multiple-length words leaves the carry indicator as set by the previous add operation.

Overflow Indicator: The overflow indicator stores the indication of overflow when the addition operation involves signed operands. The overflow indicator is set to 1 if the sign bits of the addend and augend are the same but the sign of the sum is different. The overflow indicator is set to 0 by any conditional logand that designates overflow as the condition to be tested.

P-Register: The P- (program address) register holds the program address during the interpretive mode of operation. At the logand level of operation, the P-register serves a variety of functions. During multiply and divide operations the P-register, in conjunction with the A-register, serves as a multiplier quotient register. In multiplication, the P-register contains the multiplier before the multiply logand is started and the least-significant part of the product after completion of the logand. In division, the P-register contains the least-significant part of the dividend before the divide logand is started and the quotient after completion of the logand. The P-register, in conjunction with the A-register, can be shifted left or right any number of places. The P-register

can receive inputs from either the E- or A-register, or from the address counter.

T-Register: The T- (transfer) register serves primarily as the output buffer, with the E-register handling the least-significant half of the 3Ø-bit output words. The T-register serves as the least-significant part of the input buffer, with the E-register handling 15-bit input words and the most significant part of the 30-bit input words. When not used for input or output, the T-register can serve as an auxilary storage register, receiving inputs from the E- or A-registers.

Input/Output Communication

The computer communicates with peripheral devices through I/O channels designated A, B, and C. Each channel uses a pair of cables, one input and one output cable, and the channels are often referred to as cables. Cables A and B are 3Ø-bit parallel channels, and cable C is a 15-bit parallel channel. The computer executes special input/output logands for cable selection, peripheral device selection, and data transfer in or out of the computer. An I/O counter provides sequencing and timing for I/O operations. An interrupt function is provided which allows peripheral devices to initiate the execution of program steps to effect the transfer of data.

BR-174 BUFFER-EXTENDED MEMORY (BEM)

The BEM (figure 7-31) is a buffer/extended memory unit designed for use with the Model 133C Computer. The BEM contains a 16,384 16-bit word memory, internal timing and clock circuits, data storage registers, and input/output circuits to control various input/output digital data devices. The BEM increases the memory capacity of the Model 133C, and provides for the inputting and outputting of data simultaneously with computer logic operations. The BEM contains nearly the same mechanical and electrical components as the Model 133C Computer, and therefore has nearly the same physical, electrical, and operational characteristics. The BEM is connected to the Model 133C Computer by six standard interconnecting cables which are not considered part of the BEM.

Functional Description of BEM

The BEM interfaces with and becomes an extension of the Model 133C Computer. The BEM provides additional random access core storage which can be addressed by the computer program in the same manner as the computer internal core storage. The BEM also provides four additional input/output channels each of which has interface characteristics identical to the computer cable C. Control information for the four input/output channels is stored in dedicated core locations of the BEM by the computer program. Using that control information, each channel operates independent of the computer to process data requests from peripheral devices, transferring the data to or from the BEM core storage.

BEM TIMING.—The BEM is a synchronous machine, deriving its primary timing from a 1-megahertz clock pulse. The clock generator in the BEM is synchronized to the clock oscillator in the Model 133C Computer. Selected operations are indicated by an operation counter, and executed during an operation cycle which is sequenced by a state counter.

Clock Generator.—The BEM clock pulses are generated on a clock generator card which is identical to and interchangeable with the clock generator card in the Model 133C Computer. The crystal-controlled oscillator on the computer clock generator card provides the sync input to the pulse generator portion of both cards to ensure that both machines have the same clock rate. Operating at a 1-megahertz frequency, clock pulses normally occur at 1-microsecond intervals. However, the normal clock output can be inhibited by control logic, thereby reducing the clock frequency or stopping the clock altogether.

Operation Counter.—The BEM performs only four basic operations:

a. Memory access for the Model 133C Computer.
b. Input/Output of one 15-bit data word
c. Processing Control Function (CF) logands from Model 133C Computer.
d. Processing interrupts.

The operation being performed is indicated by the state of the operation counter, and in case of simultaneous needs to perform different types of operations, the operation counter determines priority.

State Counter.–The state counter sequences the micro-operations which are selected by the operation being performed. Each operation is executed during a state counter cycle which takes from 5 to 12 microseconds, depending on the complexity of the operation. The selected operation determines the number and sequence of states in a state counter cycle, and the micro-operations of the operation involved are executed during specific states of the resulting state counter cycle.

DATA HANDLING ELEMENTS.–Data handling elements comprise those circuit elements which are used for storing, handling, and acting upon data within the BEM. The principal data handling elements include the magnetic core memory, four 15-bit registers, a 15-bit parallel half adder, and a parity generator.

Magnetic Core Memory.–The memory provides random access storage for 16,384 words. The memory is a destructive read out (DRO) memory and, therefore, requires that each read operation be followed by a write operation to restore the contents of the accessed memory location. Read and write operations are synchronized to the clock, and each operation requires 1 microsecond to be completed. Therefore, a complete memory cycle requires 2 microseconds.

All words are stored in memory with a 16th bit added for a parity check. During each write operation, the status of the 16th bit is established such that an odd number of 1's are contained in the 16-bit word stored in memory. During each read operation, the 16-bit word is examined to determine whether the number of 1's is odd or even. An even number of 1's sets a parity error indicator.

Address Register.–During a memory operation, the desired memory address is held in the R-register. The contents of the R-register are decoded by address selection circuits to select one of the 16,384 storage locations. The desired memory address is transferred from the computer M-register for computer access in extended memory. The desired memory address is read from a dedicated storage location for input/output operations.

Memory Exchange Register.–The F-register is the only register through which words are read from or written into core storage. During computer access in extended memory the word read can be transferred from the F-register to the computer E-register, or the word to be written can be transferred from the computer E-register to the F-register. The F-register contents can be transferred to the R-register to be used as an address, or to the X- or Y-registers to be transmitted to peripheral devices. The F-register can be loaded from any one of the four input channels, from any of the four channel interrupt lines, from the X- or Y-registers, from the control panel data switches, or from the half adder.

Half Adder.–The 15-bit parallel half adder is used as an address counter. The contents of the F-register are always inputting to the half adder, while a logically controlled carry input causes either a 1 or Ø to be added to the number in the F-register. The word out of the half adder can be transferred to the R-register, the X- or Y-registers, or back to the F-register.

NTDS I/O INTERFACE.–The BEM communicates with peripheral devices through four NTDS low speed I/O channels designable channels 1, 2, 3, and 4. Each channel uses a pair of cables, one input and one output, and the channels are often referred to as cables.

Input Channels.–Each of the four input channels is functionally identical to the computer cable C IN. Each input channel contains 15 parallel data lines and 9 parallel interrupt lines. Data words are transferred into the F-register and stored in the BEM's core storage. Interrupts from the input channel cause a type II interrupt in the Model 133C Computer. The computer must respond by executing a Control Function (CF) logand to process the

interrupt. During execution of that CF logand, the interrupt status word is stored in a dedicated location of the BEM's core storage.

Output Channels.—Each of the four output channels is functionally identical to the computer cable C OUT. Each output channel contains 15 parallel data lines. The X-register serves as the output register for channels 1 and 3, and the Y-register serves as the output register for channels 2 and 4. During a data output cycle, the X- or Y-register is loaded from the BEM's core storage. During an External Function (EF) command cycle, the X- or Y-register is loaded from the computer E-register.

Interrupts.—Interrupts in the BEM are divided into two groups. Group 1 interrupts include the interrupt lines from each input channel, BEM, parity error, and the synchronizing interrupt. Group 2 interrupts are the monitor interrupts from each channel. A monitor interrupt indicates that an active channel has completed its input or output block. In the computer interrupt status word, a Group 1 interrupt sets bit 14 while a Group 2 interrupt sets bit 15.

Channel Priority.—When two or more channels simultaneously request the same operation, channel priority is established by the priority counter. The determination of priority depends on the operations presently in progress and the setting of the group priority switch on the control panel.

OPERATING CONTROLS AND INDICATORS.—All manual operating controls are on either the control panel or the test and maintenance panel. Except for neon indicators on circuit cards, all indicators are on these panels.

Control Panel.—The control panel contains the switches which initiate and control all manual operations. Fifteen input switches provide a data word which can be manually transferred to the F-register and then to any other register or to core starage. Fifteen neon indicators display the binary contents of any of the four registers or the state of 15 key control flip-flops. The remaining switches and indicators control power application, channel priority, and test operations.

Test and Maintenance Panel.—The test and maintenance panel contains switches which initiate continuous memory cycling in a manual mode. These modes are useful during adjustment, checkout, and maintenance procedures. Additional maintenance aids are provided in the form of test jacks, a voltage margin switch, and a power supply meter.

POWER AND COOLING.—The BEM operates on 12Ø/2Ø8-volt, 6Ø-hertz, 3-phase input power. The input power drives an internal cooling blower, a memory heater supply, and the motor generator. No external cooling or power regulation is required.

Cooling Blower.—The internal cooling blower provides adequate cooling for normal operations (access door closed) in an ambient temperature range of Ø to 5Ø degrees centigrade.

Memory Heater Supply.—The memory heater supply provides controlled current to the memory heaters to maintain the memory internal temperature in the operating range. The memory heater supply operates on the 12Ø/2Ø8 volt input power, and maintains memory temperature in both standby and power-on modes.

Motor Generator.—Primary power for BEM power supplies is provided by a motor generator. The motor generator provides a power source of approximately 3Ø volts, 41Ø Hertz, 3 phases, which is isolated from short-duration fluctuations in the input power source.

DC Power Supplies.—The d.c. power supplies provide regulated d.c. voltage for operating all BEM circuits. The -1Ø7 volt supply provides power for neon indicators throughout the computer. The +5Ø and +34 volt d.c. power supplies are used to develop memory current. All voltages except -1Ø7 volts d.c. are used as bias and reference voltages for the logic circuits.

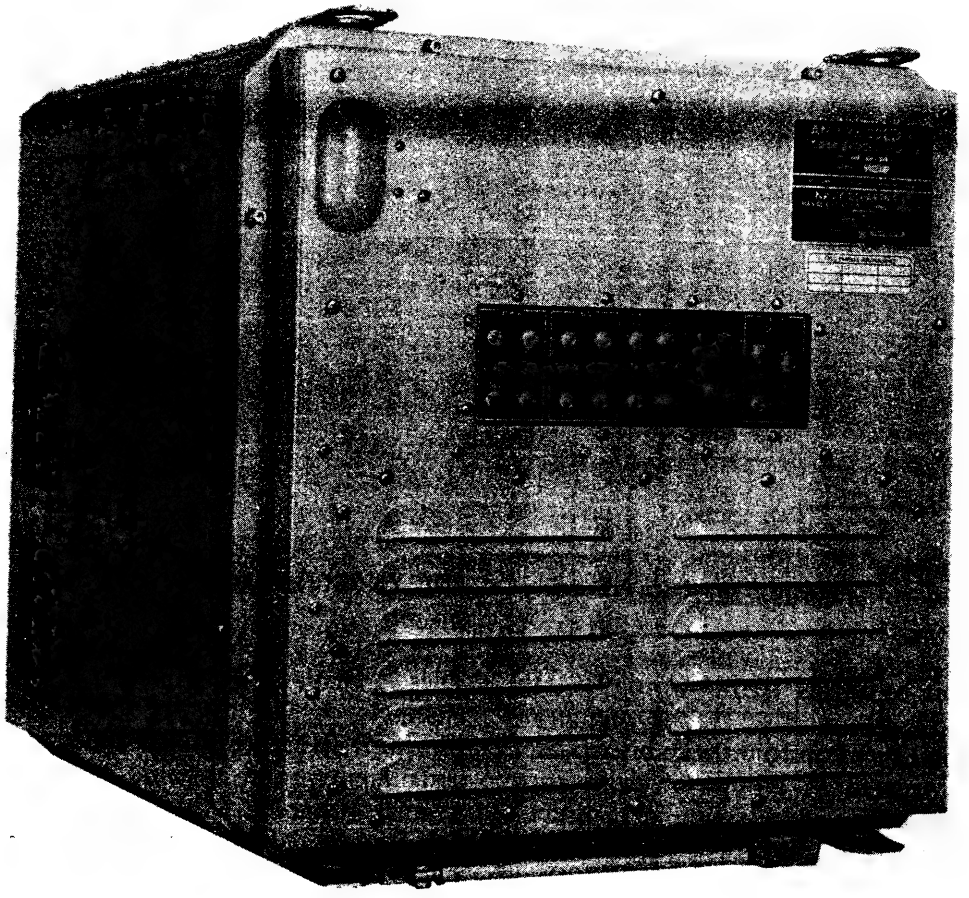
DATA PROCESSING SET AN/UYK-20(V)

The discussion which follows deals with a general purpose mini-computer, the AN/UYK-20(V) (Shown in Figure 7-37 and 7-38), and is only representative of one type of mini-computer used by the Naval Security Group.

The AN/UYK-20(V) computer is the Navy's standard mini-computer used in applications requiring a small to medium data processing capability. Due to the wide range of applications for such a computer, the Navy has established a standard software package for Government Furnished Equipment (GFE) delivery with this computer.

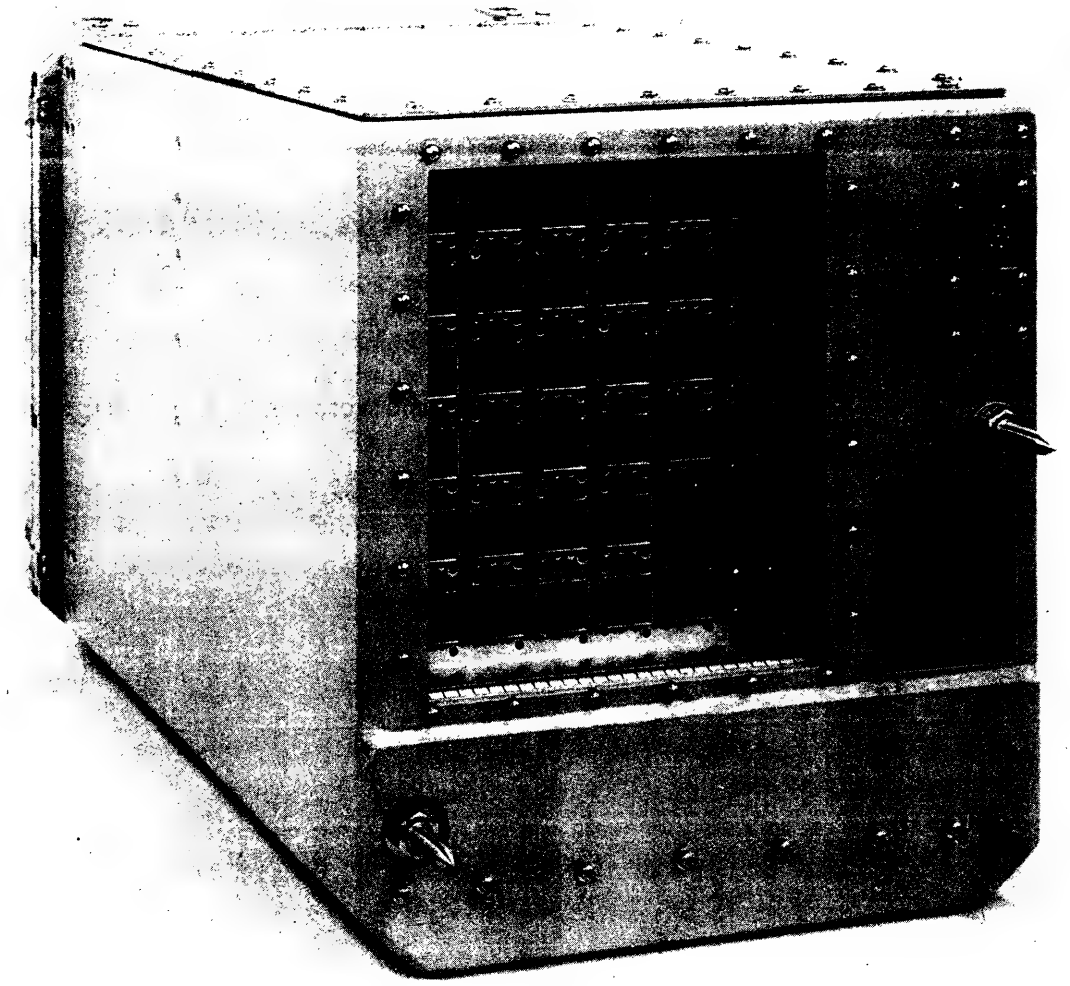
Purpose and Capabilities

The Data Processing Set, AN/UYK-20(V), is a modular medium scale, general purpose digital data processing device which uses a microprogrammed control structure. The microprogram consists of micro instructions and control data stored in read-only memory (ROM). The AN/UYK-20(V) operates from a stored program of macro instructions read from main memory to perform arithmetic operations, solve real-time problems, control other equipment and perform a variety of other data processing operations. It performs two's complement integer arithmetic operations using signed numbers. Its logic construction is parallel. The basic word length is 16 bits; which may be



1.463

Figure 7-37.—Data Processing Set, AN/GYK-20(V), Front View.



1.464

Figure 7-38.—Data Processing Set, AN/GYK-20(V), Rear View.

handled as 8 bit bytes, as 16 bit words, or as double-length 32 bit words. It has memory addressing capability of up to 65K 16 bit words; which may be treated as groups of pages for relative (vertical) addressing; or direct addressing. The main memory cycle time is 75Ø nanoseconds. The AN/UYK-20(V) communicates with peripheral equipment through an Input/Output Controller (IOC) which contains up to 16 full duplex channels. These channels may be parallel or serial channels, or a mixture of both. It has an interrupt structure that is dependent on priority assignments, and permits interruption of the normal program sequence to perform special functions. It allocates a portion of micro-memory for a user defined microprogram. It has a real-time clock and a monitor clock which operate either from an internal oscillator or from an external clock input.

Functional Characteristics

The AN/UYK-20(V) is organized around a microprogrammed controller and a two-bus data exchange structure (a 16 bit SOURCE bus and a 16 bit DESTINATION bus). The various functional elements accept bit configurations from the SOURCE bus, interpret and manipulate them, and when appropriate, return bit-configured information to the DESTINATION bus for acceptance by another functional element. The second or DESTINATION bus provides an additional communication path between the arithmetic and logic units and the various registers allowing the system to overlap functions. Control signals between sections do not use the busses, but are wired directly.

A simplified Functional Block Diagram is shown in Figure 7-39. The processor/emulator performs the arithmetic and data processing operations as directed by a program of instructions. The Input/Output (I/O) circuits transfer data between the AN/UYK-20(V) and peripheral equipment. The main memory stores instructions, operands and other data. The processor, I/O, and memory interface circuits are under control of a microprogrammed controller (MPC), which operates from its own microprogram stored in a read-only memory (ROM).

The Microprogrammed Controller (MPC) provides all control functions that enable the AN/UYK-20(V) to execute the program stored in main memory. The MPC uses and executes its own microprogram stored in a ROM to provide control functions and data manipulations. All registers and logic networks in the AN/UYK-20(V) are addressable by the micro-instructions. For each macro-instruction read from main memory, several micro-instructions are used by the MPC to provide control, timing and data transfers necessary to emulate the macro-instructions.

The processor/emulator contains logic circuits which augment the MPC. It contains an instruction register to hold the macro-instruction during emulate; and other registers and subsections that operate under MPC control to form an efficient, general purpose processor.

The memory interface handles the transfer of information between the processor or the MPC and main memory, and between I/O control and main memory. The memory interface is asynchronous, using requests and acknowledges.

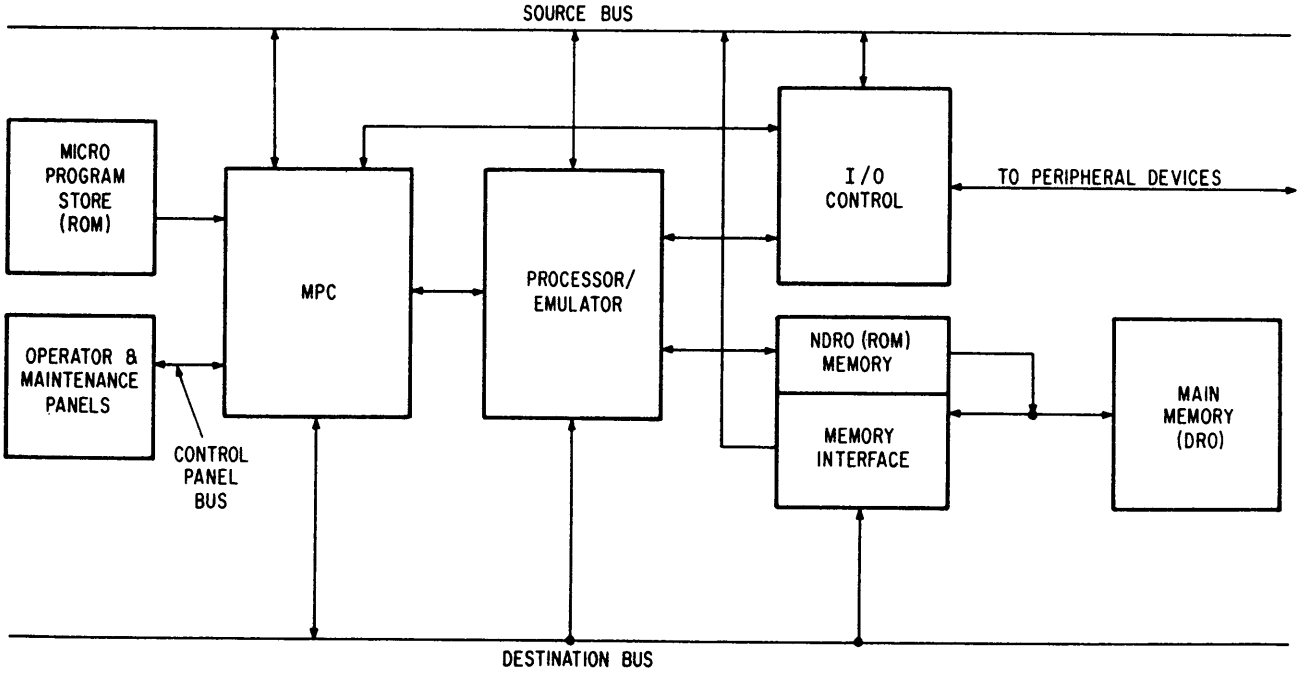


Figure 7-39.—Functional Block Diagram of the Data Processing Set AN/GYK-20(V). 1.465

The Main Memory provides storage for the macroprogram. It is available in 8192 (8k) word increments to a maximum of 65,536 words (65k).

A 192 word non-destructive readout (NDRO) semi-conductor memory provides the capability to load user programs into the main memory. The particular NDRO bootstrap instructions vary according to the channel and type of peripheral device used for main program loading. The manufacturer pre-programs the NDRO bootstrap memory at the time of manufacture on an individual NDRO card, and the bootstrap program can only be altered by substituting a different NDRO card.

The I/O control section provides for communication between the AN/UYK-20(V) and peripheral equipments, including up to 16 I/O channels. The system provides asynchronous parallel I/O channels expandable in groups of four channels; and/or synchronous or asynchronous serial channels which are expandable in groups of two channels. The serial and parallel channels can be combined in any combination up to 16 as long as the parallel channels are provided in groups of 4, and the serial channels are provided in groups of 2 or 4. The channels within a group must have identical interface characteristics. All channels are fully duplexed to permit simultaneous input and output transmission.

The computer design permits the addition of a direct memory access (DMA) capability. Two memory busses provide the interfaces. One bus connects the memory with the processor, while the second bus interfaces the memory with a user-provided external controller. The processing for memory bus remains unchanged.

Physical Characteristics

The AN/UYK-20 (V) (4ØØ Hz configuration) and the AN/UYK-20X (V) (6ØHz configuration) both consist of a single cabinet. A hinged door, called the Control-Monitor, forms the front of the cabinet. Its front side contains an operator control panel with the most essential controls and indicators. A more complete maintenance control panel is mounted on the back side of the door and is accessible with the door open. An alarm horn, air intake grill and filter are also on this door. Immediately behind the door is the memory chassis, which is hinged and mounted on slides so it may be extended and completely exposed for servicing. Behind the memory, and accessible when the memory chassis is extended, are the processor/IOC chassis, and the power supply. Figure 7-40 shows the AN/UYK-20 (V) with hinged doors open. The rear panel of the cabinet contains the power connector and grounding stud. As you face the cabinet, the left side contains air exhaust grills for the power supply, and processor/IOC chassis, each of which has an associated blower. The memory chassis has an associated blower but no external exhaust grill. Provision has been made in the base for a free-standing mount, as well as mounting within a standard 19-inch rack.

The processor/IOC chassis is called the Processor-Verifier Unit. It contains two sizes of printed circuit cards. The processor circuits are contained on single-width cards and most of the I/O circuits are contained on triple-width cards. The single-width circuit cards are mostly single-layer boards with printed wiring on both sides; the triple-width cards are three-layer boards. Single width cards have one 56-pin connector; triple width cards have two connectors.

The rear panel of the processor/IOC chassis contains the I/O connectors. With the chassis removed from the cabinet, the rear panel can be opened for access to the I/O connector wire wrap and the circuit card connector wire wrap. The processor/IOC chassis contains a Program Kit, Micro Memory MK-1723 (V)/UYK-20 (V) and may contain one or more of the various interface kits which are available, depending on the I/O types and interface levels required. Each Program Kit consists of a set of circuit cards.

The memory is made up of three different types of circuit boards. These are the Memory Control Board (MCB), the Memory Data Board (MDB) and the Core Memory Unit generally called the Memory Array Board (MAB). Each MCB contains control and addressing circuits for up to 32k of memory; each MDB contains a data register and bit drivers for up to 32k of memory;

1.466

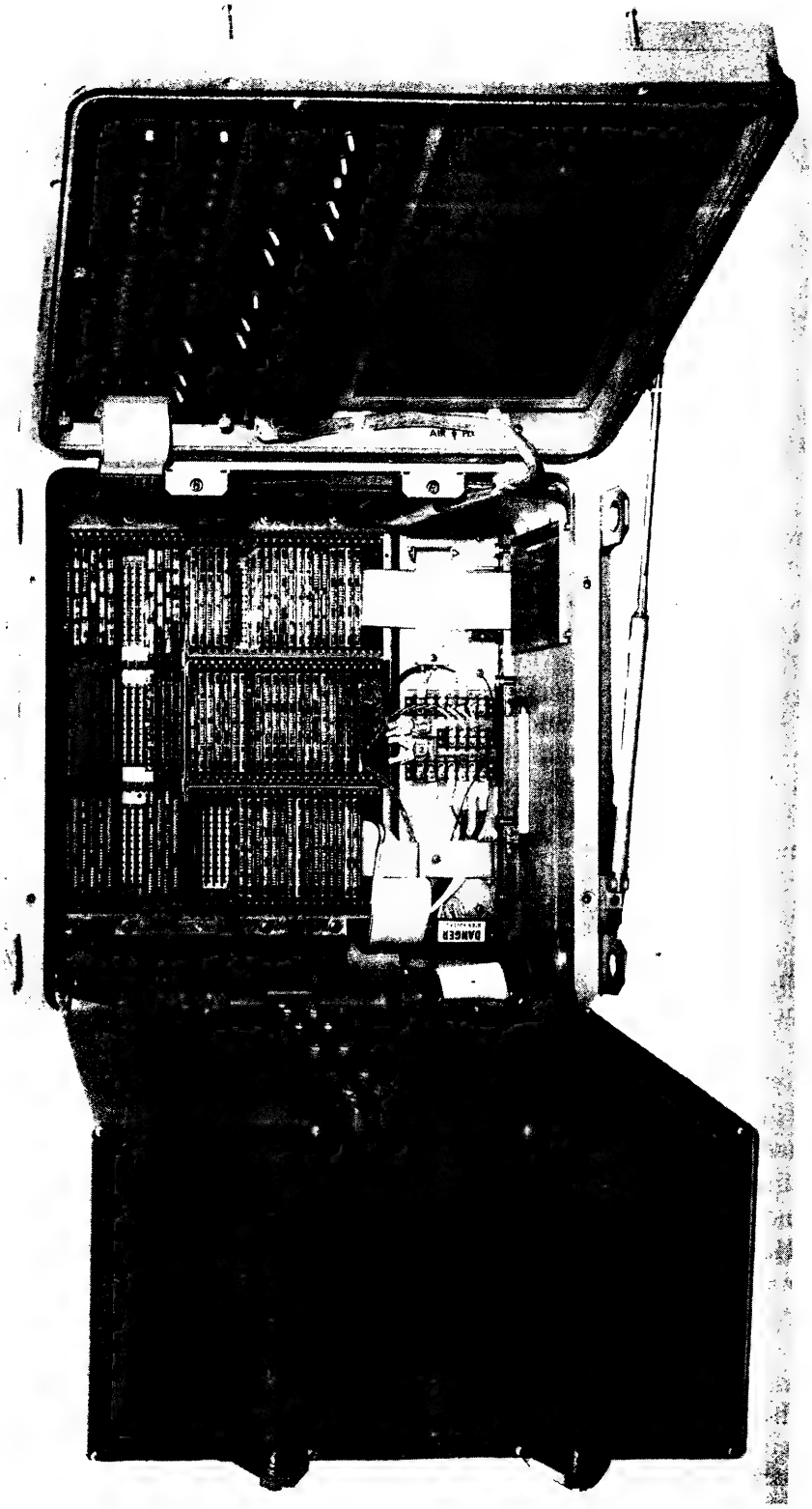


Figure 7-40.—Data Processing Set, AN/GYK-20(V), Opened- ut.

and each MAB contains an 8k X 16k bit core matrix, associated drives, and sense amplifiers. The memory chassis with two MCB and two MDB boards is designated the Core Memory-Control Unit. From one to eight Memory Array Boards (MAB) are inserted into the chassis to provide the required memory capacity; from 8k to 65k words.

The power supply is a single chassis and supplies all DC power needed by the AN/UYK-20 (V). The power supply exists in six configurations, depending on the input power requirements.

AN/UYK-20(V)
Maintenance Concepts

The maintenance concept for the AN/UYK-20 (V) requires that all maintenance actions be performed by the technicians at the installation level. The equipment design allows for modular construction which utilizes a repair or throwaway concept of plug-in printed circuit cards and modules at the organizational maintenance level. For ease of maintenance, the diagnostic software and firmware routines which are supplied with each computer are used as the primary maintenance/failure isolation tool. Failures not automatically isolated by the diagnostic software will be isolated via manual procedures. These procedures require standard test equipment, technical manual instructions, and wiring and timing diagrams. In addition, a built-in maintenance equipment panel is incorporated in the design for use with the software routines to monitor, detect and isolate failures. The maintenance concept allows all preventive and primary corrective maintenance tasks to be performed at the organizational maintenance level, and requires no intermediate level maintenance. However, a limited number of modules are designated for depot repair.

Organizational level maintenance consists of: preventive maintenance, system operational checks, troubleshooting, replacement of defective, discardable and depot level repairable modules, and piece-part repair or replacement of defective chassis mounted parts. Preventive maintenance and system operational checks are performed in accordance with Maintenance Requirements Cards (MRC's).

PROGRAMMING

Programming is, simply, the process of planning the solution to a problem. Thus, by writing:

1. Take the reciprocal of the resistance of all resistors (expressed in ohms);
2. Sum the values obtained in step 1;
3. Take the reciprocal of the sum derived in step 2. A generalized process or program for finding the total resistance of a parallel resistance circuit has now been derived.

To progress from this example to preparing a program for a computer is not difficult; however, one basic characteristic of the computer must be kept in mind. It cannot think. It can only follow certain commands and these commands must be correctly expressed and must cover all possibilities. Thus, a program, if it is to be useful in a computer, must be broken down into specifically defined operations or steps. Then the instructions, along with other data necessary for performing these operations or steps, must be communicated to the computer in the form of a language or code that is acceptable to the machine. In broad terms, the steps that the computer follows in executing a program are as follows: It reads the instructions (sequentially unless otherwise programmed), and in accordance with these instructions it (1) locates the parameters (constants) and such other data as may be necessary for problem solution, (2) transfers the parameters and data to the point of manipulation, (3) manipulates the parameters and data in accordance with certain rules of logic, (4) stores the results of such manipulations in a specific location, and (5) provides the operator (user) with a useful output. Even in a program of elementary character such as the one above, this would involve breaking each of the steps down into a series of machine operations. Then these instructions and the parameters and data necessary for problem solution must be

translated into a language or code that the computer can accept.

Consideration of these steps indicates that programming a fairly complex problem will involve writing a large number of instructions and keeping track of the great many memory cells that are used for storage of these instructions and the data necessary for problem solution. This is time consuming and could lead to error.

To reduce the amount of time required for writing a complex program and the possibility of error, an aid called the compiler has been developed. The compiler is, itself, a program which takes certain orders and then writes, in a form the machine understands, the instructions necessary for a given computer to execute these orders. Compilers are built at various levels, or degrees of complexity. The simplest form of compiler takes one mnemonic phrase and writes one machine instruction.

A mnemonic code is an abbreviated term describing something used to assist the human memory. For example, to shift the contents of the A-register right nine places. Instead of writing "Shift the A-register right one one base eight places", the mnemonic code term "RSH.A9" is used as a substitute. This would cause the compiler to write an instruction which would shift the contents of the A-register right 11_8 places. Note the advantages: (1) no opportunity to use the wrong function code; and (2) no necessity to convert the shift count to octal.

Note: A computer written on this level is commonly referred to as an "Assembler."

A more sophisticated compiler may take a statement such as "MULTIPLY PRINCIPAL BY RATE" and generate all the instructions necessary for the computer to:

1. Locate the factors involved (in this case the principal and the rate)
2. Transfer these factors to the arithmetic unit
3. Perform the indicated arithmetic (or logical) operation (in this case the multiplication of the principal by the rate)
4. Start the resultant (which in this case will be the interest or the principal)

Further, the compiler would keep track of all memory allocations, whether being used for data or instructions.

PROGRAMMING FUNDAMENTALS

Depending on the complexity of the problem to be solved, programs may vary in length from a very few instructions to many thousands of instructions. Ultimately, the program could begin to occupy a significant portion, perhaps even an excessive portion, of computer memory. One method used to preclude this possibility is to segment the program, storing seldom used portions in an auxiliary storage and reading these portions into main memory only when they are required.

Subroutines

An important method of developing this conciseness is through the use of subroutines.

Obviously, as a program grows larger certain functions will be repeated. If the instructions necessary to perform each of these repeated functions are grouped to form subroutines, these subroutines may then be referenced by a relatively few instructions in the main program each time one of the functions is to be repeated. Thus obviating the necessity of writing into the main program all of the instructions necessary to perform a function each time it is to be repeated.

Dependent upon the function being performed, a subroutine may contain other subroutines or be part of a larger subroutine.

Executive Routines

The instructions which control access to the various subroutines are referred to as the executive routine or the main program. Dependent upon the complexity of the program,

there may also be subexecutive routines, within the executive routines.

Housekeeping is a term frequently used with subroutines. At the time of entry into a subroutine, the contents of the various addressable registers may or may not be of value. Hence, the programmer will, unless he is sure they are of no value, take steps to preserve the contents of these registers upon entry into a subroutine and restore them prior to exit. This process is termed housekeeping.

THE JUMP AND RETURN JUMP INSTRUCTIONS.—The Jump and Return Jump instructions are a special type of instruction included in the computer's repertoire to facilitate construction of executive routines. These instructions provide the computer with the means to leave the sequential execution of the main program or executive routine, execute any of the subroutines stored in its memory then return to the execution of the main program once the subroutine has been executed. This process is as follows:

Execution of a Return Jump instruction causes the address of the next instruction to be executed in the main program to be stored (usually in the entry cell of the subroutine). Then it causes the instruction contained in the second cell of the subroutine to be executed. The last instruction to be executed in a subroutine will usually be a straight Jump to the address contained in the entry cell. Since a Jump instruction specifies the address of the next instruction to be executed, the computer is then provided with a means for returning to the main program once the subroutine has been executed.

Flow Charting

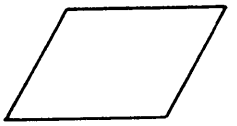
A flow diagram, or chart, serves a multitude of very important functions. As has been already noted, it is a map of how the programmer intends to solve a problem. The chart illustrates the logical steps required, the decisions to be reached, and the paths to be followed as a result of the decisions. Properly annotated it calls the programmer's attention to memory cell allocation, input/output requirements, data accuracy considerations, and register usage.

In addition, a flowchart will be of vital importance when "debugging" a program, and when making future program changes, should they be required.

LEVELS OF FLOWCHARTING.—Flowcharts, or diagrams, may be constructed at various levels or degrees of complexity. A high level flowchart usually consists of a very few symbols and presents a broad or general overview of the problem, whereas a low level flowchart may approach a one-to-one correspondence between flowchart symbols and program instructions. There will usually be several flow-charts for a given program area. These may be compared to the prints found in a maintenance manual, i.e., the block diagram to show the relationship of major units (high level), functional block diagrams showing the major circuits in a unit (intermediate level), or the schematics of the circuits (low level). Flowcharts should always be available at a level low enough to implement all the uses previously discussed.

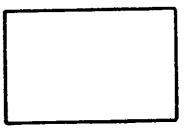
FLOWCHART SYMBOLS.—Symbols are used on a flowchart to represent the functions of an information processing system. These functions are input/output, processing, flow direction, and annotation.

INPUT/OUTPUT



This symbol represents the basic functions of entering data into the computer, or of "outputting" the data. This symbol would customarily be used for either a high level diagram or a chart where various devices are used simultaneously, since unique symbols are available for I/O processors.

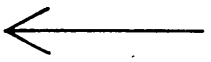
PROCESSING



The processing symbol is used for several functions. It may, at the lowest level, represent one instruction, at a higher level it will represent all of the instruction necessary to perform a given task, and the highest level

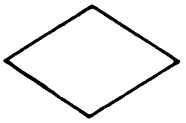
represent an entire program. In addition, when set aside from the main flow and connected to it by a dotted line, the symbol is used to contain amplifying remarks. It is then referred to as an annotation symbol.

FLOW DIRECTION



The various symbols are connected by lines; convention dicates that the flow will normally be from top to bottom and from left to right. When the conventions are followed, it is not necessary to use arrowheads unless desired, all deviations should be clearly indicated.

DECISION



The decision symbol marks the branch point in a program. Therefore, there are two possible exits from the symbol.

CONNECTOR

The connector symbol is used to identify common points in the flow paths when connecting lines cannot be drawn, or would be confusing. Usually, breaks in the main flow are annotated with letters and other common points with numbers.

TERMINAL

The Terminal Symbol represents a terminal point in a system or communication network at which data can enter or leave; e.g., start, stop, halt, delay, or interrupt.

Constructing A Flow Chart

At this point the problem previously considered will be diagramed. Certain embellishments will be added to more fully illustrate the preparation and uses of a flow diagram. Therefore, the problem is restated.

It is desired to determine the resistance of various circuits each containing as many as ten resistors in parallel. All values will be accurate to three decimal places, and the resistors values will range from 10.000 to 1000.000.

The values for all circuits will be entered at one time from a paper tape punched in teletype code. The resistances will be in decimal.

It is further stated that the program may be used as a subroutine if desired. When used as a subroutine the value found for R_t will be stored and an exit made from the subroutine. If not used as a subroutine the R_t will be typed on a teletype and the R_t for the next circuit, if any, computed. This process will continue until all circuits have been computed and typed.

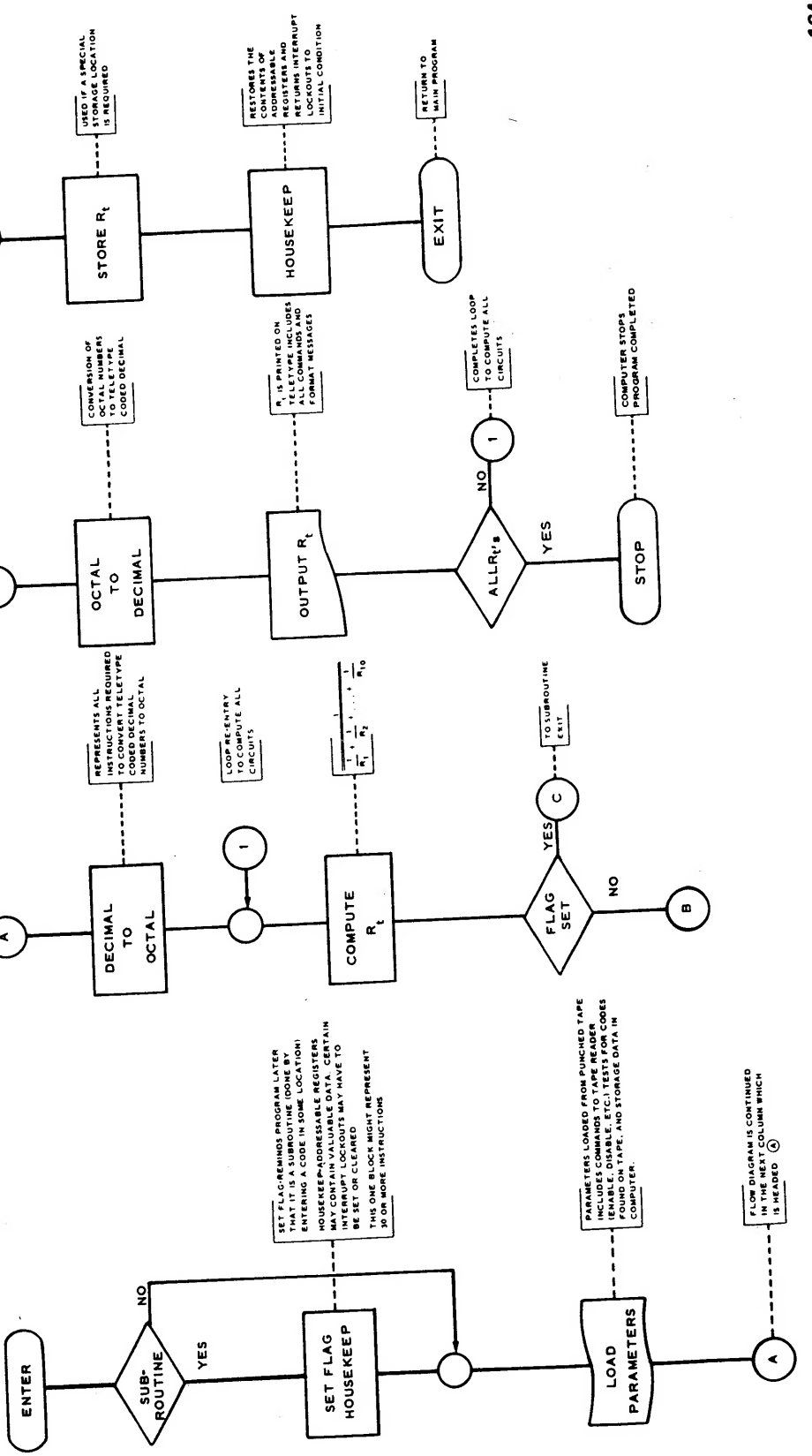
Additionally, it is assumed that if the program is used as a subroutine, the programmer does not know the contents of the various registers.

Careful study of figure 7-41 should bring to light several interesting facts. The most important observation to be made is that the actual computation will comprise a relatively small portion of the instructions. The total number of instructions could be reduced, particularly by doing the octal/decimal conversions outside the computer and by making the parameter entries into the maintenance panel and reading the solution from it. These steps would, however, largely negate the advantages of the computer. The important conclusion to be drawn from these fact is that unless a relatively simple operation must be repeated a large number of times, there is no reason to program it.

MAINTENANCE PROGRAMS

As previously discussed, a routine or program is a series of instructions which control the operations of a computer. Each instruction is used to cause some action which is part of the overall task that the computer must perform. Therefore, an instruction may be considered to be the basic building block of a computer program.

Computers can be given an overall check by means of maintenance programs. A maintenance program provides a thorough and rapid method



164.136

Figure 7-41.—Parallel Resistance Computations Flow Diagram.

for you to detect failure in a special portion of a computer. This type of overall maintenance check is very flexible and efficient. The programs use the same type tape, memory, computing, and drum circuits as operational programs. A program can be changed when the computer or auxiliary components are changed, and the program can be constantly improved. No extra test equipment is required since the computer circuits are used to perform the test. Testing by means of maintenance programs also results in the computer circuits being used in a more normal manner than during signal-tracing procedures. When a program has been checked and accepted as a good maintenance tool, it is not subject to deterioration. In contrast, test equipment may be checked and accepted only to become unreliable shortly after being placed in actual use.

Maintenance programs are divided into three main classes: reliability, diagnostic, and utility programs. Maintenance programs that are used to detect the existence of errors are called reliability programs. Reliability programs should be arranged to check as many computer circuits as possible. Maintenance programs that are used to locate the circuits in which computer malfunctions originate are called **diagnostic programs**. An effective diagnostic program should locate the source of trouble as closely as possible. Actually, in many cases reliability programs have diagnostic features, and diagnostic programs have reliability features. For convenience, a program is called either a reliability or diagnostic program depending on its intended emphasis. In general, programs that check rather than diagnose are shorter and simpler.

Basic Programs

A program is a series of instructions which control the operations of a computer. Each instruction is used to cause some action which is a part of the overall task that the computer must perform. Therefore, an instruction may be considered to be the basic building block of a computer program.

An efficient program makes full use of the instructions which are available to accomplish the task in the shortest possible time and uses the least number of instructions. In most cases, one criterion, either time or the number of instructions, has to be chosen over the other, and the program is developed along this line. If time is important, you should try to write a maintenance program which uses instructions of short duration but may use quite a few memory locations for storage. On the other hand, if time is relatively unimportant, but only a few locations are available, you must choose instructions which do a number of things or cause the computer program to run through the same program more than once.

To write a satisfactory maintenance program, it is necessary to have a thorough knowledge of the instructions that can be used. This includes execution time, the overall purpose of the instruction, when the instruction may be used, and the state of the computer after the instruction has been carried out. In addition, you should know whether the instruction can be indexed and what internal conditions must be satisfied before it can be executed.

HALT INSTRUCTION.—The halt instruction causes the computer to stop executing instructions under program control. However, any operation which is in progress at the time the halt instruction is decoded will be completed first. For example, if information is being read into the memory from a deck of punched cards, all the cards will be read before the computer halts, even if the halt instruction was issued just after the reading operation began. The address portion of the halt instruction is not used; therefore, indexing is not possible. When the computer is halted by instruction, the program counter retains the address of the instruction immediately following, so that restarting the computer will cause this next instruction to be executed.

CLEAR AND ADD INSTRUCTION.—The clear and add instruction is used to enter a quantity into the accumulators from the memory section without changing the sign or magnitude of the words. This instruction is normally used when it is desired to begin a type of addition problem. The accumulators are first cleared, and then the information selected from

the memory section by the address portion of the clear and add instruction is transferred to the computer registers. Addition of the data in the registers and the data in the accumulators is started; however, since the accumulators are cleared to +∅, this addition has the over-all effect of transferring the word from the memory section into the accumulators unchanged. The memory area used is unchanged, and the registers are cleared to +∅ after execution of the clear and add instruction.

ADD INSTRUCTION.—This instruction is similar to the clear and add instruction except that it does not provide for clearing the accumulators before the addition process begins. Thus, the add instruction will generate the sum of the data contained in the specified memory address and the data that is in the accumulators. This sum is placed in the accumulators, and the registers are cleared to +∅. It should be noted that the add instruction can cause an overflow if the numbers added together are sufficiently large. If this happens, the result in the accumulator is meaningless.

FULL STORE INSTRUCTION.—The full store instruction is used to transfer words from the accumulators into the memory area specified by the address portion of the instruction. Thus the results of any operation performed by the arithmetic section are placed in the memory section for future use. The contents of the specified register are first cleared, and then the contents of the accumulators are transferred to the memory section buffers.

Reliability Programs

Reliability programs are used in both preventive and corrective maintenance tests to detect circuit failures rapidly and to discover failures that may occur only under particular operating conditions. Examples of troubles that are not evident at all times are failures that appear at specific repetition rates or for certain combinations of bits. In order to detect such failures, it is necessary to use reliability programs which check logical operation, paths of information flow, timing, ability of the computer to perform all functions, execution of instructions, etc.

TYPES.—Reliability programs check either the logical functioning of an entire computer section or the logical functioning of individual circuit groups in a section. Whichever method is used, it is assumed that associated circuits which are not directly checked by the program are in satisfactory operating condition. Thus these programs can be considered to fall into two categories, first order and second order. First-order reliability programs check the operation of an entire computer section, while second-order programs check the operation of assemblies or circuit groups, such as registers, counters, etc. In most cases, first-order programs are merely a combination of several second-order programs.

INTERPRETATION.—A reliability program provides a good-or-bad indication regarding the ability of the tested computer section or circuit to perform its operating functions. For example, consider a reliability program that checks the switching time of relays within a specific section of a computer. As long as the switching time of the relays is within normal limits, the reliability program will indicate satisfactory operation. When the switching time is excessive however, there is an indication that maintenance is required. If the program runs successfully, there are no failures within the checked area. In the event of a failure indication, the failure may be in the area being checked or in another area that has been assumed free of trouble. Diagnostic maintenance programs should then be used to locate the source of trouble.

Diagnostic Programs

To be efficient, maintenance programs for diagnostic applications must enable you to narrow the area of a failure down to the smallest possible number of circuits. This can be accomplished by employing increasing-area, decreasing-area, overlapping-area, and large-area checks. The most effective method will depend on the particular type of computer being tested.

INCREASING-AREA CHECK.—A maintenance program using the increasing-area

check initially tests a small number of circuits. If a check indicates that all tested circuits are operating properly, successive checks are run in which progressively greater numbers of circuits are added. By this method, circuits which are found to be operating correctly are used to check other circuits. This process is continued until all circuits that can be checked by a maintenance program have been tested.

DECREASING-AREA CHECK.—When this method is used to find a trouble, a large number of circuits are initially checked by the maintenance program. If trouble is detected in a large area, additional checks are made of successively smaller portions of the equipment until the stages affected by the failure are not included in the test area. You should then be able to determine which stages are defective. If the check of a large area reveals no error, the remaining large areas of the equipment are checked until the trouble is detected. In many cases, trouble can be located more rapidly by this procedure than by the increasing-area method.

OVERLAPPING-AREA CHECK.—Another efficient method of locating trouble to within a small section of the equipment is the overlapping-area check. The routines of this type of maintenance program overlap each other. Thus, a failure is located at the overlapping portions of the routines which indicate the presence of trouble.

LARGE-AREA CHECK.—You may not be able to program an effective maintenance test for some small sections of a computer. A maintenance program can then be used only to detect the general area in which the malfunction occurs. When the general area is located, conventional trouble shooting will be necessary to find the circuit in which a failure has occurred.

Utility Programs

Utility programs are used as aids for both operation and maintenance programming procedures. This type of program is used to print out information from magnetic cores, magnetic drums, or other storage devices within the computer memory section. It is also used to transfer maintenance programs from punched cards or magnetic tape into the computer memory section. Utility tracing programs provide a printed record of the contents of various computer registers to enable you to follow maintenance program operations.

MARGINAL CHECKING.—Marginal checking is a preventive-maintenance technique that is used for some Navy and commercial computer equipment to detect the decrease in reliability of circuit parts due to aging. Aging circuit parts almost invariably change in value, current-handling capabilities, or voltage limitations. Generally, the changes brought about by aging are gradual, and you will not notice any variation in the normal operation of the equipment. For maximum equipment reliability, parts that are beginning to deteriorate must be detected and replaced before a failure occurs.

Marginal checking is usually controlled by a maintenance program. The program directs the computer to perform the normal computer operations of addition, subtraction, etc, while the program varies certain circuit parameters about their normal values. In this way, the computer is made to perform normal functions under controlled adverse operating conditions.

The amount of variation, from the normal value, that can be introduced before equipment failure occurs is called the margin of reliability of the circuit or group of circuits being tested. If the margin is regularly checked and its gradual decrease recorded; the time of circuit failure can be anticipated.